(19)日本国特許庁(JP)

# (12)特許公報(B2)

(11)特許番号
特許第7586839号
(P7586839)

(45)発行日　令和6年11月19日(2024.11.19)

(24)登録日　令和6年11月11日(2024.11.11)

(51)国際特許分類
*Ｇ１６Ｂ　30/00　(2019.01)*

ＦＩ
Ｇ１６Ｂ　30/00

請求項の数　28　（全95頁）

(21)出願番号　特願2021-566244(P2021-566244)
(86)(22)出願日　令和2年5月11日(2020.5.11)
(65)公表番号　特表2022-531790(P2022-531790A)
(43)公表日　令和4年7月11日(2022.7.11)
(86)国際出願番号　PCT/US2020/032384
(87)国際公開番号　WO2020/227718
(87)国際公開日　令和2年11月12日(2020.11.12)
審査請求日　令和5年5月8日(2023.5.8)
(31)優先権主張番号　62/845,638
(32)優先日　令和1年5月9日(2019.5.9)
(33)優先権主張国・地域又は機関
米国(US)
(31)優先権主張番号　62/860,117
(32)優先日　令和1年6月11日(2019.6.11)
最終頁に続く

(73)特許権者　519173978
カタログ テクノロジーズ， インコーポレイテッド
アメリカ合衆国 マサチューセッツ ０２１２９， チャールズタウン， メイン ストリート ５２９， スイート １２７
(74)代理人　100079108
弁理士　稲葉 良幸
(74)代理人　100109346
弁理士　大貫 敏史
(74)代理人　100117189
弁理士　江口 昭彦
(74)代理人　100134120
弁理士　内藤 和彦
(72)発明者　ロケ， ナサニエル
最終頁に続く

(54)【発明の名称】　ＤＮＡに基づくデータ記憶における探索、算出、および索引付けのためのデータ構造および動作

(57)【特許請求の範囲】

【請求項１】

核酸分子のプールに記憶されたデジタル情報からシンボルストリング内の特定の位置における特定のシンボル値のランクを取得する方法であって、各シンボルが、シンボル値およびシンボル位置を有し、前記方法が、

（ａ）前記シンボルストリングを表す識別子核酸分子の第１のプールを取得することであって、前記プールが、粉末、液体、または固体の形態を有し、前記第１のプール内の各識別子核酸分子が成分核酸分子を含み、前記成分核酸分子の少なくとも一部分が、１つまたは複数のプローブに結合するように構成される、第１のプールを取得することと、

（ｂ）前記シンボルストリングから導出されたカウンタシンボルストリングを表す識別子核酸分子の第２のプールを取得することであって、各カウンタシンボルが、前記シンボルストリングのすべてのｗ個のシンボル内の前記特定のシンボル値のランニングカウントを表す、第２のプールを取得することと、

（ｃ）（１）前記特定の位置に先行するｗ個のシンボルのすべてのブロック、または（２）前記特定の位置を含むｗ個のシンボルのブロックを含む、前記特定の位置に先行するｗ個のシンボルのすべてのブロックのいずれかに対して、前記特定のシンボル値の前記ランニングカウントを示す対応するカウンタシンボルを表す前記第２のプール内の前記識別子核酸分子を少なくとも標的とするように、第２の一連のプローブを有する（ｂ）の前記第２のプールにアクセスすることによって、第１のカウントを取得することと、

（ｄ）（１）前記特定の位置に先行しもしくは前記特定の位置を含む、（ｃ）で計数さ

れていないシンボルを表す、または（２）（ｃ）で計数されたが前記特定の位置に先行しないもしくは前記特定の位置を含まないシンボルを表す、前記第１のプール内の１つまたは複数の別個の識別子核酸分子を標的とするように、第１の一連のプローブを有する（ａ）の前記第１のプールにアクセスすることによって、第２のカウントを取得することと、
（ｅ）前記第１のカウントおよび前記第２のカウントから、前記シンボルストリング内の前記特定の位置における前記特定のシンボル値の前記ランクを取得することと
を含む方法。
【請求項２】
前記第１のプール内の前記識別子核酸分子が、識別子核酸分子の存在が、前記シンボル位置における特定のシンボル値を表すように、前記シンボルストリングにマッピングされたビットストリングを表す、請求項１に記載の方法。
【請求項３】
（ｃ）の前記第１のカウントが、前記特定の位置に先行するｗ個のシンボルのすべてのブロックを表すとき、（ｄ）の前記第１の一連のプローブが、前記特定の位置に先行しまたは前記特定の位置を含む、（ｃ）で計数されていないシンボルを表す前記第１のプール内の１つまたは複数の別個の識別子核酸分子を標的とし、前記シンボルストリング内の前記特定の位置における前記特定のシンボル値の前記ランクが、（ｅ）で前記第１および第２のカウントを合計することによって取得される、請求項１に記載の方法。
【請求項４】
（ｃ）の前記第１のカウントが、前記特定の位置を含むｗ個のシンボルの前記ブロックを含む、前記特定の位置に先行するｗ個のシンボルのすべてのブロックを表すとき、前記第１の一連のプローブが、（ｃ）で計数されたが前記特定の位置に先行しないまたは前記特定の位置を含まないシンボルを表す前記第１のプール内の１つまたは複数の別個の識別子核酸分子を標的とし、前記シンボルストリング内の前記特定の位置における前記特定のシンボル値の前記ランクが、（ｅ）で前記第１のカウントから前記第２のカウントを引くことによって取得される、請求項１に記載の方法。
【請求項５】
前記第１のカウントが、（ｃ）の前記標的とされた識別子核酸分子に対応するカウンタシンボル値を読み取ることによって取得される、請求項１に記載の方法。
【請求項６】
前記第２のカウントが、（ｄ）からの前記標的とされた識別子核酸分子を読み取ることによって取得される、請求項１に記載の方法。
【請求項７】
（ａ）の前記第１のプールが、（ｂ）の前記第２のプールである、請求項１に記載の方法。
【請求項８】
（ａ）の前記第１のプールが、（ｂ）の前記第２のプールとは別個である、請求項１に記載の方法。
【請求項９】
前記第１のプール内の対応する識別子核酸分子の存在が、第１のシンボル値を示し、前記第１のプール内の対応する識別子核酸分子の不在が、第２のシンボル値を示す、請求項１に記載の方法。
【請求項１０】
前記シンボルストリングが、長さｎを有し、前記カウンタシンボルが、ｂビットによって表され、ここでｂが$\log_2(n+1)$の上限である、請求項９に記載の方法。
【請求項１１】
前記カウンタシンボルストリングが、ｎをｗで割った上限個のカウンタシンボルを含み、ｂにｎをｗで割った上限を掛けた値に対応する長さを有するカウンタビットストリングによって表される、請求項１０に記載の方法。
【請求項１２】

前記特定の位置が、ｗ個のシンボルの第１のブロックの範囲内である場合、ｗ個のシンボルの前記第１のブロックに先行する前記ランニングカウントが０である、請求項１に記載の方法。

【請求項１３】

前記特定の位置が、ｗ個のシンボルの前記第１のブロックの範囲内にない場合、前記特定の位置に先行するｗ個のシンボルのすべてのブロックの前記カウンタシンボルが、前記シンボルストリングの０からｗ＊Ｂ（ｘ）－１の位置範囲内の前記特定のシンボル値の発生の数を表し、０が前記シンボルストリングの第１の位置であり、ｘが前記シンボルストリング内の前記特定の位置に対応し、Ｂ（ｘ）がｘをｗで割った下限である、請求項１２に記載の方法。

【請求項１４】

（ｃ）の前記第２のプール内の前記標的とされた識別子核酸分子が、位置ｂ＊Ｂ（ｘ）およびｂ＊（Ｂ（ｘ）＋１）－１によって画定される前記範囲内であり、０の位置が、前記シンボルストリング内の前記第１の位置に対応する、請求項１３に記載の方法。

【請求項１５】

前記第２のカウントが、前記シンボルストリングのｗ＊Ｂ（ｘ）からｘの前記位置範囲内の前記特定のシンボル値の発生の数に対応し、ｘが前記シンボルストリング内の前記特定の位置に対応し、位置０が、前記第１の位置であり、Ｂ（ｘ）が、ｘをｗで割った下限である、請求項１に記載の方法。

【請求項１６】

ｗが、前記シンボルストリングのｗ個のシンボルを表すビットの長さが、ビットの長さｂと同等になり、前記カウンタシンボルを表すように設定される、請求項１に記載の方法。

【請求項１７】

ｗが１の値に設定される、請求項１に記載の方法。

【請求項１８】

前記第１のカウントが、前記特定の位置を含むｗ個のシンボルの前記ブロックに対応する前記カウンタシンボルを表す（ｃ）の識別子核酸分子を標的とすることによって取得され、前記ランクが前記第１のカウントと同等である、請求項１７に記載の方法。

【請求項１９】

ステップ（ｄ）が実行されない、請求項１８に記載の方法。

【請求項２０】

前記シンボルストリング内のｗ個のシンボルのブロックが、識別子核酸分子の前記第１のプール内の隣接して順序付けられた識別子核酸分子のブロックにマッピングされる、請求項１に記載の方法。

【請求項２１】

前記シンボルストリング内の前記シンボルがビットであり、各ビットが識別子核酸分子にマッピングされ、したがって識別子核酸分子の前記第１のプール内の前記識別子の有無が、前記ビットの値を示す、請求項２０に記載の方法。

【請求項２２】

前記シンボルストリングの固定長のサブストリングが、固定数の可能な固有の識別子核酸分子からの固定数の固有の識別子核酸分子を含むコードワードにマッピングされる、請求項２１に記載の方法。

【請求項２３】

追加の情報を使用して、前記第１および第２のプールからの識別子核酸分子の書込み、アクセス、および読取りのエラーを検出および補正することをさらに含む、請求項１に記載の方法。

【請求項２４】

前記追加の情報が、前記第１および第２のプールの前記識別子核酸分子に記憶される、請求項２３に記載の方法。

【請求項２５】

前記シンボルストリングが、ビットストリングを表す、請求項１に記載の方法。
【請求項２６】
前記シンボルストリングの各シンボルが、固定数のビットに対応する、請求項２５に記載の方法。
【請求項２７】
（ｂ）の識別子核酸分子の異なる第２のプールが、特有のシンボル値のインスタンスの数を計数する異なるカウンタシンボルストリングを表し、異なる各カウンタシンボルストリングが、対応する特有のシンボル値のインスタンスを計数する、請求項２５に記載の方法。
【請求項２８】
Ｍ個の選択された成分核酸分子を物理的に組み立てることによって、識別子核酸分子が形成され、前記Ｍ個の選択された成分核酸分子の各々が、Ｍ個の異なる層に分離された別個の成分核酸分子のセットから選択される、請求項１に記載の方法。
【発明の詳細な説明】
【技術分野】
【０００１】
関連出願の相互参照
本出願は、２０１９年５月９日出願の"NUCLEIC ACID-BASED DATA STORAGE: SEARCH AND COMPUTE"という名称の米国仮特許出願第６２／８４５，６３８号、２０１９年６月１１日出願の"DATA STRUCTURES FOR DNA STORAGE"という名称の米国仮特許出願第６２／８６０，１１７号、および２０１９年８月２２日出願の"INDEX AND SEARCH OF INFORMATION STORED IN DNA"という名称の米国仮特許出願第６２／８９０，２４３号に対する優先権および利益を主張する。上記に参照した出願の内容全体が、参照により本明細書に組み込まれている。
【背景技術】
【０００２】
核酸デジタルデータ記憶は、情報を符号化し、長期間にわたって記憶するための安定した手法であり、データは、磁気テープまたはハードドライブ記憶システムより高い密度で記憶される。加えて、低温および乾燥条件で保管された核酸分子に記憶されているデジタルデータは、６０，０００年またはそれよりも長い年数の後に取り出すことができる。
【０００３】
核酸分子に記憶されたデジタルデータにアクセスするための１つの方法は、核酸分子をシークエンシングすることである。したがって、核酸デジタルデータ記憶は、頻繁にアクセスされるわけではないが、長期間にわたって記憶または格納されるべき大量の情報を有することのあるデータを記憶するための理想的な方法となりうる。
【０００４】
核酸分子にデータを記憶する既存の方法は、デジタル情報（たとえば、２進コード）を塩基ごとの核酸配列に符号化することに依拠しており、したがって、配列内の塩基間の関係が、デジタル情報（たとえば、２進コード）に直接変換される。しかし、そのような塩基ごとのデノボ核酸合成は、エラーを起こしやすい上に高価である。さらに、これらの既存の核酸デジタルデータ記憶方法を介して記憶されたデータでは、単一塩基分解能でデータが符号化されているとき、最初に分子集合全体をデジタル情報に変換させなければ、特定の機能を実行することができない。たとえば、そのような機能には、データセット内で特有の照会パターンが発生したか否か、発生の数、および各発生の場所を含めて、論理関数、加算、減算、および照会探索など、データがディスクに記憶されるときに一般に実行される基本的なタスクが含まれる。
【発明の概要】
【課題を解決するための手段】
【０００５】
本開示は、最適化されたデータ構造および関数によってＤＮＡに記憶されたデータの探

索および抽出を可能にすることを対象とする。したがって、核酸分子に記憶されたデータに対して特定の関数を実行するシステムおよび方法が、本明細書に提供される。本開示は、少なくとも、（１）核酸分子に記憶された情報への効率的なアクセスおよび探索を提供するデータ構造、（２）核酸分子に記憶された情報の正確かつ迅速な読取り、（３）核酸分子に記憶された情報のサブセットへのアクセスへの標的を絞った手法、（４）核酸分子に記憶された情報のセット内の特定のビットまたはシンボル値のカウントを判定するランク関数、（５）核酸分子に記憶された情報のメッセージにおける特有のパターンの発生の計数、位置付け、および抽出を含む関数、ならびに（６）核酸分子に記憶されたデータを分類するｉｆ－ｔｈｅｎ－ｅｌｓｅ演算という関心領域を包含する。

【０００６】

本明細書に記載するシステムおよび方法は、あらゆる核酸配列を読み取るために使用することができるが、本開示は、識別子核酸分子（本明細書では、単に「識別子」または「識別子分子」とも呼ぶ）に特定の符号化方法を使用して書かれた核酸配列に記憶された情報の読取り、アクセス、および探索を行うときに特に有利である。各識別子分子の核酸配列は、シンボルストリング（たとえば、ビットストリームまたはシンボルストリーム）内の特定のシンボル値（たとえば、２つのシンボル値「０」および「１」のみが可能であるとき、または別法として３つ以上のシンボル値が可能であるとき、１つまたは一連のビット）、そのシンボルの位置（たとえば、ランクまたはアドレス）、または両方に対応する。たとえば、識別子分子の有無は、それぞれ１または０のビット値を伝えることができる（または逆も同様である）。識別子核酸分子は、成分核酸分子（本明細書では、単に「成分」または「成分分子」とも呼ぶ）の組合せ配置を含む。成分の核酸配列は、固有のセット（層とも呼ばれる）に分離される。識別子分子は、複数の成分分子をともに結合する（または他の方法で組み立てる）ことによって組み立てられており、各成分分子は、各層から選択された配列を有する。いくつかの場合、成分分子は自己組織化して、識別子分子を形成する。可能な識別子配列のセットが、成分配列の様々な可能な組合せの組合せに対応する。たとえば、Ｃ個の成分配列がＭ個の層に分離され、ここで$c_i$が第ｉの各層内の成分配列の数を表す場合、形成することができる可能な識別子配列の数は、$c_1 \times c_2 \times \ldots \times c_M$によって表すことができる。一例として、１２個の層からなり、各層が１０個の成分配列を収容する符号化方式は、$10^{12}$個の異なる固有の識別子配列をもたらすことができる。各識別子配列がビットストリーム内の１ビットに対応する場合、この符号化方式は、１ＴＢのデータを表すことができる。

【０００７】

ＤＮＡから情報を読み取る効率を改善するための１つの方法は、データ構造を使用して、データストリングのデータブロックの場所を記憶して示すことを伴う。たとえば、大きいデータストリングは、２つまたはそれよりも多いコンテナに分離して記憶することができる。ユーザがアクセスしようとする情報をどのコンテナが収容しているかを判定するために、システムは、場所（たとえば、コンテナ番号または配置）を保持するＢ木またはトリプルストア構造にアクセスすることができる。これにより、ユーザは、データストリングを収容している各コンテナ内の情報を読み取るのではなく、自身が探していた情報に好都合にアクセスすることが可能になる。

【０００８】

データ構造はまた、ビット／シンボル値のランク付けおよびデータ内の特有のパターンの探索などの高速動作を可能にする構成で、核酸分子内にデータを記憶することができる。たとえば、システムは、データストリングにおけるビット／シンボル値の発生のランニングカウントを表すカウンタアレイにアクセスすることができる。これにより、システムは、システム内のすべての単一の核酸分子を読み取ることなく、ビット／シンボル値のランクを判定することが可能になる。個々の各核酸分子または配列を読み取る必要なく、データ内の特有のパターンの効率的な探索を可能にする圧縮に好適な形式でデータを記憶するために、Ｂｕｒｒｏｗｓ－Ｗｈｅｅｌｅｒ変換を介して、または接尾辞アレイに、データを変換することができる。

【０００９】
いくつかの態様では、複数のブロックを取得することによって、デジタル情報を核酸分子に記憶するシステムおよび方法が本明細書に提供されており、各ブロックはシンボルストリングを含み、ブロックＩＤに関連付けられる。そのようなシステムおよび方法は、記憶されたデータまたは関連付けられた核酸分子を特徴付ける情報を記憶または暗示することによって、記憶されたデータの探索を容易にするために、ブロックＩＤを使用することができることから、利点を提供する。このシステムおよび方法では、ブロックがコンテナに割り当てられ、コンテナに関連付けられるべき複数の識別子核酸配列にマッピングされ、各識別子核酸配列は成分核酸配列を含んでおり、成分核酸配列の少なくとも一部分は、１つまたは複数のプローブに結合するように構成される。システムおよび方法は、複数の識別子核酸配列の個々の識別子核酸分子をさらに構築し、割り当てられたコンテナ内に個々の識別子分子を記憶し、コンテナおよびそのコンテナに関連付けられた複数の識別子核酸配列の識別情報を含む物理アドレスが、関連付けられたブロックＩＤを使用して判定されるように構成される。
【００１０】
たとえば、ブロックＩＤは、整数、ストリング、トリプル、属性リスト、または意味注釈である。いくつかの実装形態では、物理アドレスは、関連付けられたブロックＩＤを使用して物理アドレスにアクセスすることを容易にするように設計されたデータ構造に記憶される。データ構造は、Ｂ木、トライ、またはアレイのうちの１つとすることができる。データ構造の少なくとも一部分は、デジタル情報とともに索引内に記憶することができ、索引は、第２のコンテナに関連付けられた第２の複数の識別子核酸配列を含むことができる。索引は、ノードを有するＢ木またはトライデータ構造とすることができ、ノードは各々、第２の複数の識別子核酸配列のうちの別個の複数の識別子核酸配列に対応する。第１のノードを含む別個の識別子にアクセスすることができ、それに続いてそれらの識別子に関連付けられた値を読み取ることができ、これらのステップを後続のノードに対して繰り返すことができ、ブロックＩＤを使用して、第１のノードの値に関する後続のノードを含む別個の複数の識別子の識別情報を判定することができる。第１のノードは、Ｂ木またはトライの根ノードとすることができ、プロセスは、葉ノードの値が読み取られるまで継続して、ブロックＩＤに対するブロックが存在するかどうか、および対応する物理アドレスが何であるかを示すことができる。ブロックＩＤは、シンボルストリングとすることができ、Ｂ木またはトライデータ構造の各ノードは、シンボルストリングの可能な接頭辞に対応する。葉ノード自体は、葉ノードによって指定されたシンボルストリングに一致するブロックＩＤに関連付けられた物理アドレスを表すことができる。
【００１１】
いくつかの実装形態では、データ構造はアレイであり、アレイの各要素は、第２の複数の識別子核酸配列のうちの別個の複数の識別子核酸配列に対応する。アレイの各要素は、ブロックＩＤに対応することができ、または関連付けられたブロックＩＤの物理アドレスを記憶することができる。いくつかの実装形態では、物理アドレスは、物理アドレスを追加のデータ構造に記憶する必要なく、ブロックＩＤが物理アドレスにマッピングするように、ブロックＩＤに固有に構成される。ブロックＩＤは、物理アドレスに関連付けられた複数の識別子核酸配列のうちのすべての識別子核酸配列によって共用される複数の成分核酸配列にマッピングすることができる。いくつかの実装形態では、ブロックに関連付けられた複数の識別子核酸配列は、隣接して順序付けられた識別子核酸配列を含み、したがって前記複数の識別子核酸配列は、前記識別子範囲内の第１および最後の識別子核酸分子の識別情報を含む識別子範囲によって、対応する物理アドレスに指定される。前記識別子範囲内の第１および最後の識別子核酸配列は、整数によって表すことができる。
【００１２】
いくつかの実装形態では、ブロックＩＤは、親シンボルストリングの対応するブロックによって表されるシンボルストリング内の位置である。親シンボルストリングは、別のシンボルストリングにおけるパターンの発生を計数または位置付けするためのデータ構造に

よって表すことができる。適当なデータ構造には、Ｂｕｒｒｏｗｓ－Ｗｈｅｅｌｅｒ変換、接尾辞アレイ、接尾辞木、および転置索引が含まれる。データ構造がアレイである場合、アレイの各要素は、ブロックＩＤに対応する別個の複数の識別子を含むことができる。アレイの各要素は、関連付けられたブロックＩＤの物理アドレスを記憶することができる。

【００１３】

いくつかの実装形態では、物理アドレスは、第２の複数の識別子配列を含む識別子のプールから、一連のプローブを使用してアクセスされる。データ構造は、磁気記憶デバイス、光記憶デバイス、フラッシュメモリデバイス、またはクラウドストレージに記憶することができる。一連のプローブを使用して、ブロックに関連付けられた識別子にアクセスすることができる。プローブは、ＰＣＲプライマーまたは親和性タグ付きオリゴヌクレオチドとすることができ、それぞれアクセスのためにＰＣＲまたは親和性プルダウンアッセイを使用する。

【００１４】

いくつかの態様では、核酸分子のプールに記憶されたデジタル情報からビットストリングまたはシンボルストリング内の特定の位置における特定のビットまたはシンボル値のランクを取得するシステムおよび方法が本明細書に提供されており、各ビットまたはシンボルは、値および位置を有する。システムおよび方法は、ビットストリングまたはシンボルストリングを表す識別子核酸分子の第１のプールを取得することであって、プールが、粉末、液体、または固体の形態を有し、第１のプール内の各識別子核酸分子が成分核酸分子を含み、成分核酸分子の少なくとも一部分が、１つまたは複数のプローブに結合するように構成される、第１のプールを取得することと、ビットストリングまたはシンボルストリングから導出されたカウンタシンボルストリングを表す識別子核酸分子の第２のプールを取得することであって、各カウンタシンボルが、ビットストリングまたはシンボルストリングのすべてのｗ個のビットまたはシンボル内の特定のビットまたはシンボル値のランニングカウントを表す、第２のプールを取得することとを伴う。システムおよび方法は、少なくとも（１）特定の位置に先行するｗ個のビットもしくはシンボルのすべてのブロック、または（２）特定の位置を含むｗ個のビットもしくはシンボルのブロックを含む、特定の位置に先行するｗ個のビットもしくはシンボルのすべてのブロックのいずれかに対して、特定のビットまたはシンボル値のランニングカウントを示す対応するカウンタシンボルを表す第２のプール内の識別子核酸分子を標的とするように、第２の一連のプローブを有する第２のプールにアクセスすることによって、第１のカウントを取得することをさらに伴う。（１）特定の位置に先行しもしくは特定の位置を含む、第１のカウントで計数されていないビットもしくはシンボルを表す、または（２）第１のカウントで計数されたが特定の位置に先行しないもしくは特定の位置を含まないビットもしくはシンボルを表す、第１のプール内の１つまたは複数の別個の識別子核酸分子を標的とするように、第１の一連のプローブを有する第１のプールにアクセスすることによって、第２のカウントが取得される。第１のカウントおよび第２のカウントから、ストリング内の特定の位置における特定のビットまたはシンボル値のランクが取得される。

【００１５】

本明細書に記載するランク関数の１つの利点は、ストリング全体を読み取る必要がないことである。代わりに、ランク関数は、選択的アクセス演算を使用して、特有のカウントを提供する核酸分子のサブセットを読み取る。このようにして、本明細書に記載するランク演算は、記憶されたすべての核酸分子の完全な読取りを必要とするブルートフォース演算と比較すると、より時間効率的かつ高い費用効果で実行される。

【００１６】

識別子は、Ｍ個の選択された成分核酸分子を物理的に組み立てることによって形成することができ、Ｍ個の選択された成分核酸分子の各々は、Ｍ個の異なる層に分離された別個の成分核酸分子のセットから選択される。

【００１７】

いくつかの実装形態では、第１のカウントが、特定のビットに先行するｗ個のビットの

すべてのブロックを表すとき、第２のカウント内の第１の一連のプローブは、特定のビットに先行しまたは特定のビットを含む、第１のカウントで計数されていないビットを表す第１のプール内の１つまたは複数の別個の識別子核酸分子を標的とし、ビットストリング内の特定のビットのランクは、第１および第２のカウントを合計することによって取得される。いくつかの実装形態では、第１のカウント内の第１のカウントが、特定のビットを含むｗ個のビットのブロックを含む、特定のビットに先行するｗ個のビットのすべてのブロックを表すとき、第１の一連のプローブは、第１のカウントで計数されたが特定のビットに先行しないまたは特定のビットを含まないビットを表す第１のプール内の１つまたは複数の別個の識別子核酸分子を標的とし、ビットストリング内の特定のビットのランクは、第１のカウントから第２のカウントを引くことによって取得される。

【００１８】

いくつかの実装形態では、第１のプールが第２のプールであり、または第１のプールおよび第２のプールが別個である。第１のカウントおよび第２のカウントは、それぞれ第２のプールおよび第１のプールから、標的とされた識別子分子を読み取ることによって取得することができる。第１のカウントの場合、標的とされた識別子に対応するカウンタシンボル値を読み取りまたは判定することができる。ストリング内のｗ個のビットまたはシンボルのブロックは、第１のプール内の隣接して順序付けられた識別子核酸分子のブロックにマッピングすることができる。たとえば、第１のプール内の識別子分子の有無は、ストリング内の特定の値に直接相関しない。ストリングの固定長のサブストリング（ワード）を、固定数の可能な固有の識別子核酸分子からの固定数の固有の識別子核酸分子を含むコードワードにマッピングすることができる。識別子に記憶された追加の情報を使用して、識別子の書込み、アクセス、および読取りにおけるエラーを検出および補正することができる。

【００１９】

各カウンタシンボルは、ｂ個のカウンタビットのストリングによって表すことができる。いくつかの実装形態では、ストリングは、ｎ個のビットまたはシンボルの長さを有し、ｂは$\log_2(n+1)$の上限である。カウンタシンボルストリングは、ｎをｗで割った上限個のカウンタシンボルを含むことができ、ｂにｎをｗで割った上限を掛けた値に対応する長さを有するカウンタビットストリングによって表される。いくつかの実装形態では、特定のビットまたはシンボルが、ｗ個のビットまたはシンボルの第１のブロックの範囲内である場合、ｗ個のビットの第１のブロックに先行するランニングカウントは０である。ｗの値は、ｂの値または１に設定することができる。たとえば、第１のカウントは、特定のビットまたはシンボルを含むｗ個のビットまたはシンボルのブロックに対応するカウンタシンボルを表す第２のプール内の識別子核酸分子を標的とすることによって取得され、ランクは、第１のカウントと同等である。この例では、最後の第２のカウントを取得する必要はない。

【００２０】

いくつかの実装形態では、特定のビットが、ｗ個のビットまたはシンボルの第１のブロック内にない場合、特定のビットまたはシンボルに先行するｗ個のビットまたはシンボルのすべてのブロックのカウンタシンボルは、０からｗ＊Ｂ（ｘ）－１の位置範囲内の特定の値を有するストリング内のビットまたはシンボルの数を表し、０はストリングの第１の位置であり、ｘはストリング内の特定のビットまたはシンボルの位置に対応し、Ｂ（ｘ）はｘをｗで割った下限である。たとえば、第１のカウントに対する第２のプール内の標的とされた識別子核酸分子は、位置ｂ＊Ｂ（ｘ）およびｂ＊（Ｂ（ｘ）＋１）－１によって画定される範囲内であり、０の位置は、ストリング内の第１の位置に対応する。第２のカウントは、位置範囲ｗ＊Ｂ（ｘ）～ｘ内に特定の値を有するストリング内のビットまたはシンボルの数に対応することができ、ここで、ｘはストリング内の特定のビットまたはシンボルの位置に対応し、位置０は第１の位置であり、Ｂ（ｘ）はｘをｗで割った下限である。

【００２１】

ストリングがビットストリングであるいくつかの実装形態では、第１のプール内の識別子核酸分子は、ビットストリングを表し、したがって識別子の存在は、ビット位置のビット値「１」を表す。ビットストリングは、シンボルストリングを表すことができ、シンボルストリング内の特定のシンボルに対してランクが取得される。

【００２２】

ストリングがシンボルストリングであるいくつかの実装形態では、第１のプール内の対応する識別子核酸分子の存在は、第１のシンボル値を示し、第１のプール内の対応する識別子核酸分子の不在は、第２のシンボル値を示す。シンボル値は、シンボル値のセット（たとえば、アルファベット）から選択することができ、カウンタシンボルストリングは、特定のシンボル値を有するシンボルの数のランニングカウントを示す。シンボルストリングは、ビットストリングを表すことができる。たとえば、ストリング内の各シンボルは、固定数のビットに対応する。いくつかの実装形態では、識別子核酸分子の異なる第２のプールは、特有のシンボル値のインスタンスの数を計数する異なるカウンタシンボルストリングを表し、異なる各カウンタシンボルストリングが、対応する特有のシンボル値のインスタンスを計数する。

【００２３】

いくつかの態様では、核酸分子のプールからデジタル情報をフェッチするシステムおよび方法が本明細書に提供される。システムおよび方法は、識別子核酸分子の第１のプールを取得することを伴い、プールが、粉末、液体、または固体の形態を有し、第１のプール内の各識別子核酸分子が成分核酸分子を含み、成分核酸分子の少なくとも一部分が、１つまたは複数のプローブに結合するように構成され、識別子核酸分子は、シンボルストリングを表し、したがってシンボル値は、前記第１のプール内の対応する識別子核酸分子の有無によって示される。第１の一連のプローブによって前記第１のプールにアクセスして、前記第１のプールからの識別子核酸分子のサブセットを有する第２のプールを作成し、第１の一連のプローブの各々は、成分核酸分子のうちの少なくとも１つを標的とする。前記第２のプールからの識別子核酸分子の前記サブセットの配列を読み取り、前記配列を使用して、少なくともシンボルストリング内のシンボルのサブセットを取得する。

【００２４】

各識別子核酸分子は、Ｍ個の層の各々からの成分核酸配列を有する別個の成分核酸分子を含み、各層は、成分核酸配列のセットを含む。Ｍ個の層は、論理的に順序付けることができ、各層の成分核酸配列を論理的に順序付けることができる。いくつかの実装形態では、識別子核酸分子は、識別子核酸配列を第１の層内の対応する成分核酸配列によって分類し、識別子核酸配列を第２の層内の対応する成分核酸配列によって細分し、残りのＭ－２個の層の各々に対して細分プロセスを繰り返すことによって、論理的に順序付けられた識別子核酸配列に対応する。各識別子配列は、照会木内のパスとして表される一連の成分核酸配列を含むことができ、照会木は、根ノードから開始し、各層に１つのインスタンスでＭ個のインスタンスにわたって分岐し、葉ノードで終了し、各葉ノードは、識別子核酸配列を表す。したがって、第１の一連のプローブは、照会木内の根ノードからの部分パスまたはフルパスに対応することができる。フルパスは、Ｍ個のプローブを含む根から葉までのパスに対応することができ、したがって一連のプローブは、単一の識別子核酸分子を標的とし、部分パスは、Ｍ個より少ないプローブに対応し、したがって一連のプローブは、異なる配列を有する識別子核酸分子の複数の母集団を標的とする。異なる配列を有する識別子核酸分子の複数の母集団は、少なくとも第Ｍの層内の異なる成分核酸分子に対応することができる。

【００２５】

いくつかの実装形態では、第１のプールは、複数の一連のプローブによってアクセスされる。たとえば、システムおよび方法は、第１のプールを少なくとも２つの複製プールに分割し、手法のステップを、一連のプローブの各々を有する前記複製プールの各々で実行することを実行する。第１のプールは、少なくとも２つの複製プールに分割する前に複製することができる（たとえば、ＰＣＲを介して）。いくつかの実装形態では、プローブの

サブシリーズを有する識別子核酸分子の第１のプールにアクセスして、識別子核酸分子の中間プールを作成する。プローブの後続のサブシリーズを有する中間プールにアクセスして、識別子核酸分子の第２の中間プールまたは識別子核酸分子の第２のプールを形成することができる。少なくとも２つの中間プールを組み合わせて、別の中間プールまたは第２のプールを形成することができる。プローブは、ＰＣＲプライマー（ＰＣＲを介したアクセスの場合）、または親和性タグ付きオリゴヌクレオチド（親和性プルダウンアッセイを介したアクセスの場合）とすることができる。

【００２６】

いくつかの態様では、長さｎのビットストリングを含むメッセージ内の長さｐの特定のビットパターンのカウントを取得するシステムおよび方法が本明細書に提供される。システムおよび方法は、ビットストリングＬを表す識別子核酸分子の第１のプールを取得し、ビットストリングＬは、メッセージのＢｕｒｒｏｗｓ－Ｗｈｅｅｌｅｒ変換行列の最後の列であり、第１のプールは、粉末、液体、または固体の形態を有し、第１のプール内の各識別子核酸分子は、成分核酸分子を含み、成分核酸分子の少なくとも一部分は、１つまたは複数のプローブに結合するように構成される。ビットストリングＬから導出されたカウンタシンボルストリングを表す識別子核酸分子の第２のプールが取得され、各カウンタシンボルは、特有のビット値「１」を有するビットストリングＬ内のすべてのｗ個のビットに対するビットの数のランニングカウントを示すｂ個のカウンタビットのストリングによって表される。一連のプローブを使用して、ビットストリングＬ内のビット値「１」の発生の総数に対するカウンタシンボルを表す第２のプールからの識別子核酸分子にアクセスする。第２のプールからアクセスされた識別子核酸分子を読み取って、ビットストリングＬ内のビット値「１」の発生の総数を計数する。各ビット値の発生の総数は、メッセージのＢｕｒｒｏｗｓ－Ｗｈｅｅｌｅｒ変換行列の第１の列Ｆを再構築するためのものである。第１の列Ｆ内の第ｐのビット値の範囲を画定する第１の位置ｈおよび最後の位置ｚが、ｈおよびｚを含めて判定される。第１の一連のプローブを使用して、第１のプールおよび第２のプールからの識別子核酸分子にアクセスして、Ｌ内の位置ｈ－１におけるパターンの第（ｐ－ｉ）のビット値のランク$r_{h-1}$を計算し、ここでｉ＝１である。第２の一連のプローブを使用して、第１のプールおよび第２のプールからの識別子核酸分子にアクセスして、Ｌ内の位置ｚにおけるパターンの第（ｐ－ｉ）のビット値のランク$r_z$を計算する。

【００２７】

$r_{h-1}$が$r_z$に等しい場合、メッセージ内のパターンの発生のカウントは０として設定される。そうではなく、$r_{h-1}$が$r_z$に等しくない場合、ｈは、Ｆ内の第（ｐ－ｉ）のビット値の第（$r_{h-1}$＋１）のインスタンスの索引に設定される。ｚは、Ｆ内の第（ｐ－ｉ）のビット値の第$r_z$のインスタンスの索引に設定される。ループカウンタｉが１だけ増分され、アクセスおよび索引付けステップが、ｉ＝ｐ－１になるまで複数回繰り返される。メッセージ内のパターンの発生のカウントは、ｚ－ｈ＋１として計数される。本手法によって提供される少なくとも１つの利点は、データセット内のすべての単一のビットを読み取ることなく、大きいデータセットにわたって探索を実行することができることであり、代わりに本開示は、選択的アクセスおよびランク演算を介して、パターンの発生を論理的に位置付ける。本明細書に提供される別の利点は、実行時間を最小にし、処理量を増大させるために、アクセスおよびランク演算を第１および第２のプール上で並列に実行することができることである。

【００２８】

第１のプールは、第２のプールと同じにすることができ、または第２のプールとは別個にすることができる。いくつかの実装形態では、メッセージのＢｕｒｒｏｗｓ－Ｗｈｅｅｌｅｒ変換から導出された接尾辞アレイを表す識別子核酸分子の第３のプールが取得され、接尾辞アレイの各要素は、メッセージ内のＬの対応する要素の位置を示す少なくとも$log_2$（ｎ）個のビットのビットストリングによって表される。システムおよび方法は、カウントが０より大きいとき、ｈおよびｚに対する最終値を含むｈとｚと間の位置における接尾辞アレイ内の要素に対応する第３のプール内の識別子核酸分子にアクセスすること

によって、メッセージ内のパターンの発生をさらに位置付けることができる。いくつかの実装形態では、メッセージを表す識別子核酸分子の第４のプールが取得される。システムおよび方法は、前記第１の場所および第１の場所を取り囲む位置の近傍に対応する第４のプール内の識別子核酸分子にアクセスすることによって、パターンの第１の場所のコンテキストをさらに抽出することができる。

【００２９】

いくつかの実装形態では、第１のプール内の対応する識別子核酸分子の存在は、ビット値１を示し、第１のプール内の対応する識別子核酸分子の不在は、ビット値０を示す。ｂは、ｌｏｇ２（ｎ＋１）の上限に等しくすることができる。カウンタシンボルストリングは、ｎをｗで割った上限個のカウンタシンボルを含むことができ、ｂにｎをｗで割った上限を掛けた値に対応する長さを有するカウンタビットストリングによって表される。Ｌ内のｗ個のビットの第１のブロックに先行するあらゆるビット値のランニングカウントは０である。ｗは、ｂの値または１に設定することができる。システムおよび方法は、ビットストリングＬ内のビットのブロックを、第１のプール内の隣接して順序付けられた識別子核酸分子のブロックにマッピングすることができる。ビットストリングＬの固定長のサブストリングを、固有の識別子核酸分子の固定サイズのセットから選択された固定数の固有の識別子核酸分子によって表されるコードワードにマッピングすることができる。追加の情報を使用して、第１および第２のプールからの識別子核酸分子の書込み、アクセス、および読取りのエラーを検出および補正し、追加の情報を識別子に記憶することができる。

【００３０】

いくつかの態様では、メッセージ内の長さｐの特定のビットパターンのカウントを取得するシステムおよび方法が本明細書に提供される。システムおよび方法は、ビットストリングＬを表す識別子核酸分子の第１のプールを取得し、ビットストリングＬは、メッセージのＢｕｒｒｏｗｓ－Ｗｈｅｅｌｅｒ変換行列の最後の列であり、第１のプールは、粉末、液体、または固体の形態を有し、第１のプール内の各識別子核酸分子は、成分核酸分子を含み、成分核酸分子の少なくとも一部分は、１つまたは複数のプローブに結合するように構成される。特有のビット値を有するビットの数のランニングカウントを表す、ビットストリングＬから導出されたカウンタシンボルストリングを表す識別子核酸分子の第２のプールが取得される。メッセージ内の特定のビットパターンのカウントは、第１のプールおよび第２のプールからの識別子核酸分子に選択的にアクセスすることによって取得される。この技法によって提供される少なくとも１つの利点は、データセット内のすべての単一のビットを読み取ることなく、大きいデータセットにわたって探索を実行することができることであり、代わりに本技法は、選択的なアクセスを介してパターンの発生を論理的に位置付けることを伴う。本明細書に提供される別の利点は、実行時間を最小にし、処理量を増大させるために、アクセス演算を第１および第２のプール上で並列に実行することができることである。

【００３１】

いくつかの態様では、長さｎｓのシンボルストリングを含むメッセージ内の長さ$p_s$の特定のシンボルパターンのカウントを取得するシステムおよび方法が本明細書に提供され、各シンボルは、ｒ個のシンボル値のセットから選択される。システムおよび方法は、メッセージのＢｕｒｒｏｗｓ－Ｗｈｅｅｌｅｒ変換行列の最後の列であるシンボルストリングＬを表す識別子核酸分子の第１のプールを取得し、第１のプールは、粉末、液体、または固体の形態を有し、第１のプール内の各識別子核酸分子は、成分核酸分子を含み、成分核酸分子の少なくとも一部分は、１つまたは複数のプローブに結合するように構成される。システムおよび方法は、識別子核酸分子のｒ個の第２のプールをさらに取得し、ｒ個の第２のプールの各々は、Ｌから導出されたカウンタシンボルストリング$C_v$に対応し、ここでｖ＝１、２、．．．、ｒであり、これは対応するシンボル値$R_v$を有するＬ内のシンボルの数のランニングカウントを表す。システムおよび方法は、第１のプールおよびｒ個の第２のプールからの識別子核酸分子に選択的にアクセスすることによって、メッセージ内の長さ$p_s$の特定のシンボルパターンのカウントを取得する。この手法によって提供さ

れる少なくとも１つの利点は、データセット内のすべての単一のビットを読み取ることなく、大きいデータセットにわたって探索を実行することができることであり、代わりに本手法は、選択的なアクセスを介してパターンの発生を論理的に位置付ける。本明細書に提供される別の利点は、実行時間を最小にし、処理量を増大させるために、アクセス演算を第１および第２のプール上で並列に実行することができることである。

【００３２】

いくつかの実装形態では、本手法は、Ｂｕｒｒｏｗｓ－Ｗｈｅｅｌｅｒ変換行列の第１の列Ｆを再構築することを含む。一連のプローブを使用して、Ｌ内の対応する各シンボル値$Ｒ_{v}$の発生の総数を表すｒ個の第２のプールの各々における最後のカウンタシンボルから、識別子核酸分子にアクセスすることができ、対応する各シンボル値$Ｒ_{v}$の発生の前記総数を使用して、Ｆを再構築することができる。パターン内の第ｐのシンボル値を有するＦ内の位置の範囲を判定することができる。いくつかの実装形態では、本手法は、一連のプローブを使用して、第１のプールおよび対応する第２のプールから識別子核酸分子にアクセスして、範囲にすぐ先行する位置におけるＬ内の対応するシンボル値の第１のランク、および範囲の末端の位置におけるＬ内の対応するシンボル値の第２のランクを判定することと、第１のランクおよび第２のランクを使用して、この範囲を、パターン内の後続のシンボルに先行する対応するシンボルのインスタンスを有するＦ内の位置の範囲に更新することとを含む。

【００３３】

いくつかの実装形態では、第１のランク$ｒ_{ｈ－１}$は、Ｌ内の位置ｈ－１におけるパターン内のそれぞれの先行するシンボル値であり、第２のランク$ｒ_{ｚ}$は、Ｌ内の位置ｚにおけるパターン内のそれぞれの先行するシンボル値である。メッセージ内のパターンの発生のカウントは、第１および第２のランクの最終値に基づいて判定することができる。たとえば、カウントは、第１および第２のランクの最終値間の差である。第１および第２のランクが等しいと判定された場合、カウントを０に設定することができる。

【００３４】

いくつかの実装形態では、本手法は、メッセージのＢｕｒｒｏｗｓ－Ｗｈｅｅｌｅｒ変換から導出された接尾辞アレイＳＡを表す識別子核酸分子のプール、ＳＡプールを取得することを含み、ＳＡの各要素は、メッセージ内のＬの対応する要素の位置を示す少なくとも$ｌｏｇ_{2}$（ｎ）個のビットのビットストリングによって表される。方法およびシステムは、カウントが０より大きいと仮定して、Ｆ内の位置の最終範囲によって与えられる位置におけるＳＡの要素に対応するＳＡプール内の識別子核酸分子にアクセスすることによって、メッセージ内のパターンの発生を位置付けることができる。本手法は、メッセージを表す識別子のメッセージプールを取得することを含むことができ、前記第１の場所および第１の場所を取り囲む位置の近傍に対応するメッセージプール内の識別子核酸分子にアクセスすることによって、パターンの第１の場所のコンテキストの抽出を可能にする。

【００３５】

いくつかの実装形態では、識別子核酸分子の第１のプールは、ｒ個の第１のプールのうちの１つであり、ｒ個の第１のプールの各々は、ビットストリング$Ｌ_{v}$に対応し、ここでｖ＝１、２、．．．、ｒであり、したがって$Ｌ_{v}$の要素は、シンボル値$Ｒ_{v}$に一致するＬの要素に対してビット値「１」、そうでない場合はビット値「０」を有し、または逆も同様である。たとえば、$Ｌ_{v}$に対応する第１のプールが、パターン内のシンボル値$Ｒ_{v}$の第１および第２のランクを判定するために使用される。

【００３６】

いくつかの態様では、核酸分子に記憶されたデジタル情報に対して動作するシステムおよび方法が本明細書に提供される。システムおよび方法は、識別子核酸分子の第１のプールを取得し、プールは、粉末、液体、または固体の形態を有し、第１のプール内の各識別子核酸分子は、成分核酸分子を含み、成分核酸分子の少なくとも一部分は、１つまたは複数のプローブに結合するように構成され、識別子核酸分子は、入力シンボルストリングを表す。第１のプール内の識別子核酸分子でｉｆ－ｔｈｅｎ－ｅｌｓｅ演算が実行され、ｉ

ｆ－ｔｈｅｎ－ｅｌｓｅ演算は、プローブを有する成分核酸分子のうちの少なくとも１つを標的とし、前記第１のプールからの識別子核酸分子のサブセットを有する中間プールを作成する。ｉｆ－ｔｈｅｎ－ｅｌｓｅ演算が繰り返され、出力シンボルストリングの少なくとも一部分を表す識別子核酸分子の最終プールが作成されるまで、すべての後続のステップにおいて中間プールが第１のプールに取って代わる。

【００３７】

この手法を使用することで、条件付きプログラムを書き込むことが可能になる。たとえば、各「ｉｆ」演算は、１つまたは複数の識別子の有無を試験し、その有無に応じて、「ｔｈｅｎ」または「ｅｌｓｅ」の分岐へ進む。この演算は、複数の条件および対応する分岐を含むことができる。演算のすべての分岐から、出力を作り出すことができる。この手法により、複数の識別子ライブラリ内の識別子のすべて（たとえば、テラビット規模）を並列に演算することが可能になる。たとえば、ライブラリが数十億のデータオブジェクトを符号化する場合、各オブジェクトを調べて出力を作り出す複素関数を、ＤＮＡに基づくプログラムとして設計し、ライブラリ上で並列に実行することができる。

【００３８】

本開示は、所望のプログラムの実行のために識別子ライブラリを複数の入力識別子ライブラリに再配置するビットをシフト、コピー、および移動する方策を含む。物理的に、１つの入力ライブラリを２つの出力ライブラリに変換する反応において、各ｉｆ－ｔｈｅｎ－ｅｌｓｅ演算を行うことができ、それらのライブラリ内のすべての識別子にわたって多重化することができる。たとえば流体伝達によって、１つの演算の出力ライブラリを別の演算の入力ライブラリへ向けることができる。ＲＡＭおよび処理力によって制限される従来のハードウェアとは異なり、本明細書に記載するＤＮＡプラットホームは、大量の入力データオブジェクトにわたって同時に低電力でプログラムを実行することが可能である。

【００３９】

識別子は、Ｍ個の層の各々から別個の成分を含むことができ、各層は成分のセットを含む。プローブは、ＰＣＲプライマーまたは親和性タグ付きオリゴヌクレオチドとすることができ、それによってそれぞれＰＣＲまたは親和性プルダウンアッセイを介してアクセスが実行される。いくつかの実装形態では、ｉｆ－ｔｈｅｎ－ｅｌｓｅ演算は、特有の成分核酸分子を含むプール内の識別子核酸分子にアクセスすることを含む。演算は、第１のプール、中間プール、または最終プールのうちの少なくとも１つを少なくとも２つの複製プールに分割または複製し（たとえば、ＰＣＲを介して）、少なくとも２つの中間プールを組み合わせて、新しい中間プールもしくは第２のプールまたは両方を形成することを含むことができる。１つまたは複数のプール上で並列に、２つまたはそれよりも多いｉｆ－ｔｈｅｎ－ｅｌｓｅ演算を実行することができる。

【００４０】

いくつかの態様では、本明細書に記載する方法のいずれかを実行するように構成されたシステムが本明細書に提供される。システムは、基質上の個別の場所（たとえば、反応コンパートメント）にＤＮＡ成分を分注し、結合反応に最適の条件を提供する試薬を分注し、ライブラリを含むＤＮＡ識別子のすべてをプールするように構成されたプリンタフィニッシャシステムとすることができる。システムは、コンテナ内に核酸分子を記憶して操作することができる（たとえば、自動化された液体の取扱いを介して）。システムは、コンパートメントまたはコンテナ内へプローブを分注して、核酸分子のサブセットにアクセスすることができる。システムは、核酸分子のプールを等分および複製するように構成することができる。

【００４１】

いくつかの態様では、本明細書に記載する方法のいずれかによるデジタル情報を表す核酸分子を含む組成物が本明細書に提供される。組成物は、成分核酸分子を含む識別子核酸分子を含む。識別子核酸分子は、プール内に収集してデジタル情報にマッピングすることができる。たとえば、識別子の存在は、シンボルストリング内の特定のビットまたはシンボル値を示し、識別子の不在は、シンボルストリング内の別のビットまたはシンボル値を

示す。
【００４２】
本開示の上記およびその他の特徴は本開示の性質およびその様々な利点を含めて、添付の図面と併せて以下の詳細な説明を考慮するとより明らかになる。
【図面の簡単な説明】
【００４３】
【図１】図１は、核酸配列に記憶されたデジタル情報の符号化、書込み、アクセス、照会、読取り、および復号を行うためのプロセスの概略を概略的に示す図である。
【００４４】
【図２】図２Ａおよび図２Ｂは、オブジェクトまたは識別子（たとえば、核酸分子）を使用して「データアットアドレス」と呼ばれるデジタルデータを符号化する例示的な方法を概略的に示す図であり、図２Ａは、ランクオブジェクト（またはアドレスオブジェクト）をバイト値オブジェクト（またはデータオブジェクト）と組み合わせて識別子を作成することを示し、図２Ｂは、ランクオブジェクトおよびバイト値オブジェクト自体が他のオブジェクトの組合せ連結であるデータアットアドレス方法の実装形態を示す。
【００４５】
【図３】図３Ａおよび図３Ｂは、オブジェクトまたは識別子（たとえば、核酸配列）を使用してデジタル情報を符号化する例示的な方法を概略的に示す図であり、図３Ａは、ランクオブジェクトを識別子として使用してデジタル情報を符号化することを示し、図３Ｂは、アドレスオブジェクト自体が他のオブジェクトの組合せ連結である符号化方法の実装形態を示す。
【００４６】
【図４】図４は、所与のサイズ（等高線）の情報を記憶するように構築することができる可能な識別子の組合せ空間（Ｃ、ｘ軸）と識別子の平均数（ｋ、ｙ軸）との間の関係の対数空間の等高線図を示す図である。
【００４７】
【図５】図５は、核酸配列（たとえば、デオキシリボ核酸）に情報を書き込むための方法の概略を概略的に示す図である。
【００４８】
【図６】図６Ａおよび図６Ｂは、別個の成分（たとえば、核酸配列）を組合せにより組み立てることによって、識別子（たとえば、核酸分子）を構築するための「積方式」と呼ばれる例示的な方法を示す図であり、図６Ａは、積方式を使用して構築された識別子のアーキテクチャを示し、図６Ｂは、積方式を使用して構築することができる識別子の組合せ空間の一例を示す。
【００４９】
【図７】図７は、成分（たとえば、核酸配列）から識別子（たとえば、核酸分子）を構築するためのオーバーラップ伸長ポリメラーゼ連鎖反応の使用を概略的に示す図である。
【００５０】
【図８】図８は、成分（たとえば、核酸配列）から識別子（たとえば、核酸分子）を構築するための付着末端結合の使用を概略的に示す図である。
【００５１】
【図９】図９は、成分（たとえば、核酸配列）から識別子（たとえば、核酸分子）を構築するためのリコンビナーゼアセンブリの使用を概略的に示す図である。
【００５２】
【図１０Ａ】図１０Ａおよび図１０Ｂは、鋳型指向結合を実証する図であり、図１０Ａは、成分（たとえば、核酸配列）から識別子（たとえば、核酸分子）を構築するための鋳型指向結合の使用を概略的に示す図であり、図１０Ｂは、１つのプールされた鋳型指向結合反応において６つの核酸配列（たとえば、成分）から組合せにより各々組み立てられた２５６個の別個の核酸配列のコピー数（存在度）のヒストグラムを示す。
【図１０Ｂ】図１０Ａおよび図１０Ｂは、鋳型指向結合を実証する図であり、図１０Ａは

、成分（たとえば、核酸配列）から識別子（たとえば、核酸分子）を構築するための鋳型指向結合の使用を概略的に示す図であり、図１０Ｂは、１つのプールされた鋳型指向結合反応において６つの核酸配列（たとえば、成分）から組合せにより各々組み立てられた２５６個の別個の核酸配列のコピー数（存在度）のヒストグラムを示す。

【００５３】

【図１１】図１１は、識別子のトライによるコンパートメント内への溶液の分注の図である。

【００５４】

【図１２】図１２は、層によって組織化されたオペレーティングシステムのシステム図である。

【００５５】

【図１３】図１３は、３つの層を有する層状のプロダクトコンストラクタおよび８つの成分からなる成分ライブラリのシステム図である。

【００５６】

【図１４】図１４は、アーカイブ演算のシステム図である。

【００５７】

【図１５】図１５は、コンテナ内へのデータのブロックの記憶の流れ図である。

【００５８】

【図１６】図１６は、ストリングのＢｕｒｒｏｗｓ－Ｗｈｅｅｌｅｒ変換を示す図である。

【００５９】

【図１７】図１７は、Ｂｕｒｒｏｗｓ－Ｗｈｅｅｌｅｒ行列に対する接尾辞アレイを示す図である。

【００６０】

【図１８】図１８は、核酸に記憶されたＢｕｒｒｏｗｓ－Ｗｈｅｅｌｅｒ変換の図例である。

【００６１】

【図１９】図１９は、核酸に記憶されたカウンタアレイの図例である。

【００６２】

【図２０】図２０は、照会木の図例である。

【００６３】

【図２１】図２１は、２分木における識別子の隣接範囲の分解の図例である。

【００６４】

【図２２】図２２は、照会指向非循環グラフへの照会木の変換の図例である。

【００６５】

【図２３】図２３は、２進ストリングに対するランク演算の実行の図例である。

【００６６】

【図２４】図２４は、ランク演算の実行のための流れ図である。

【００６７】

【図２５】図２５は、フェッチ演算の実行のための流れ図である。

【００６８】

【図２６】図２６は、２進ストリングに対するカウント演算の実行のための流れ図である。

【００６９】

【図２７】図２７は、２進ストリングに対するカウント演算のための流れ図である。

【００７０】

【図２８】図２８は、図２９の方法のステップの実行のための流れ図である。

【００７１】

【図２９】図２９は、任意のストリングに対するカウント演算の実行のための流れ図である。

【００７２】

【図３０】図３０は、ＬＦマッピングのための流れ図である。

【００７３】
【図３１】図３１は、Ｂｕｒｒｏｗｓ－Ｗｈｅｅｌｅｒ変換からのストリングの再構築のための流れ図である。
【００７４】
【図３２】図３２は、カウント演算のための流れ図である。
【００７５】
【図３３】図３３は、ｉｆ－ｔｈｅｎ－ｅｌｓｅ演算のための流れ図である。
【発明を実施するための形態】
【００７６】
本明細書に記載するアセンブリおよび方法の全体的な理解を提供するために、特定の例示的な実装形態について説明する。本明細書に記載する実装形態および特徴は、成分核酸分子から構成された識別子核酸分子におけるデータ記憶に関して具体的に説明されているが、以下に概説するすべての態様および他の特徴は、任意の好適な形で互いに組み合わせることができ、ＤＮＡに基づくデータ記憶の他の形式にも適合および適用することができることが理解されよう。
【００７７】
デジタル情報を符号化するための核酸の塩基ごとの合成は、概してすべての新しい情報記憶要求に対して別個の核酸配列の塩基ごとのデノボ合成（たとえば、ホスホロアミダイト合成）を必要とすることから、高価でありかつ時間がかかる可能性がある。本開示は、塩基ごとの合成またはデノボ合成に依拠するのではなく、代わりに成分（または成分核酸配列）の組合せ配置を含む複数の識別子または核酸配列内にデジタル情報を符号化するシステムおよび方法に関する。このようにして、本開示のシステムおよび方法は、デジタル情報記憶の効率および製品化の可能性を改善する。
【００７８】
本開示は、第１の情報記憶要求に対して別個の核酸配列（または成分）の第１のセットを作り出し、その後同じ核酸配列（または成分）を後続の情報記憶要求に対して再利用することができる方法を説明する。これらの手法は、情報からＤＮＡへの符号化および書込みプロセスにおける核酸配列のデノボ合成の役割を低減させることによって、ＤＮＡに基づく情報記憶のコストを大幅に低減させる。
【００７９】
さらに、各伸長核酸への各塩基の循環的な送達を使用するホスホロアミダイトの化学的性質または鋳型のないポリメラーゼに基づく核酸伸長などの塩基ごとの合成の実装形態とは異なり、成分からの識別子構造を使用して情報からＤＮＡに書き込むことに関する本開示のシステムおよび方法は、非常に並列化可能なプロセスであり、必ずしも循環的な核酸伸長を使用しない。したがって、本開示は、他の方法と比較すると、ＤＮＡにデジタル情報を書き込む速度を増大させる。デジタル情報を核酸分子に書き込む様々なシステムおよび方法は、各々が全体として参照により本明細書に組み込まれている、２０１７年１２月２１日出願の"NUCLEIC ACID-BASED DATA STORAGE"という名称の米国特許第１０，６５０，３１２号（ＤＮＡにおけるデジタル情報の符号化について記載）、２０１９年５月１６日に出願され、米国特許出願公開第２０１９／０３６２８１４号として公開された"SYSTEMS FOR NUCLEIC ACID-BASED DATA STORAGE"という名称の米国特許出願第１６／４６１，７７４号（ＤＮＡに基づくデータ記憶のための符号化方式について記載）、２０１９年5月１６日出願の"COMPOSITIONS AND METHODS FOR NUCLEIC ACID-BASED DATA STORAGE"という名称の米国特許出願第１６／４１４，７５８号、および２０１９年８月５日出願の"SYSTEMS AND METHODS FOR STORING AND READING NUCLEIC ACID-BASED DATA WITH ERROR PROTECTION"という名称の米国特許出願第１６／５３２，０７７号（ＤＮＡ符号化のためのデータ構造ならびにエラーの保護および補正について記載）に記載されている。
【００８０】
以下の説明は、データを核酸分子で符号化する様々なシステムおよび方法の概略から始

まり、図１～図１０に関連して説明するように、デジタルデータを符号化する核酸分子を印刷および記憶するように構成された様々な書込みおよび格納システムについて説明する。本開示は次いで、図１１～図１４に関連して様々な符号化方法について説明する。本開示は、上記で参照した特許出願に記載されている核酸分子におけるデジタル情報の書込みおよび読取りを行う方法の改善に関する。具体的には、データ構造を使用してデジタル情報を表すことで、デジタル情報の特有の特性を画定し、ＤＮＡにおけるアクセスを容易にするようにそれらの特性を組織化することによって、ＤＮＡから情報を読み取る効率を改善することができる。

【００８１】

一例として、大きいデータストリングが２つまたはそれよりも多いサブストリングに分離され、各サブストリングは核酸分子の別個のプールに記憶され、このプールは独自のコンテナ内へ配置される。どのコンテナが所望の情報を収容しているかを判定するための１つの方法は、場所（たとえば、コンテナ数または配置）を識別するデータ構造（たとえば、Ｂ木またはトライ構造など）にアクセスすることである。データ構造を使用してどのコンテナが関連するかを参照することによって、ユーザは、各コンテナ内の情報を１つずつ読み取ってからその情報が関連するかどうかを判定するのではなく、所望の情報を収容しているコンテナだけにアクセスすることができる。これにより、核酸分子内のデータストリングにおいて関連する情報にアクセスする効率が改善される。

【００８２】

上記の例の延長として、ユーザがアクセスしたい関連する情報は、コンテナのプール内の核酸分子のサブセットのみから表すことができ、または計算可能とすることができる。この場合、本開示は、核酸分子のプール全体に記憶された情報のすべてにアクセスする必要なく、関連する分子の特有のサブセットのみにアクセスする方法を提供する。そうすることで、効率が増大され、コストが低減されるはずである。プール内の核酸分子の望ましいサブセットのみにアクセスするための１つの方法は、プール内の核酸分子の特有のサブセットを標的とするために使用することができる情報を記憶しているデータ構造（たとえば、Ｂ木またはトライ構造など）を参照することである。プール内の（または異なるコンテナ内の異なるプールにわたる）核酸分子の特有のサブセットにアクセスしてそれを探索するために使用することができる特有のデータ構造の例は、図１５～図１９に関連して説明する。さらに、本開示は、探索、位置付け、および抽出機能など、核酸分子に記憶されたデータに対して特定の演算を効率的に実行するためにデータ構造に依拠するシステムおよび方法に関する。具体的には、核酸分子に記憶された特有のデータ部分にアクセスして、読取り、アクセス、およびランク付けなどの演算を実行するために１つまたは複数のデータ構造に依拠する例示的なシステムおよび方法が、図２０～図２５に関連して説明されており、核酸分子に記憶された特有のデータ部分にアクセスして、核酸分子に記憶されたデータからの特有のパターンまたは照会の探索、位置付け、および抽出などの演算を実行するための１つまたは複数のデータ構造に依拠するシステムおよび方法が、図２６～図３２に関連して説明されている。最後に、論理ｉｆ－ｔｈｅｎ－ｅｌｓｅ演算が、図３３に関連して説明されている。

【００８３】

概して、本開示は、データ（１もしく０ビットのストリング、またはシンボルストリングによって表され、各シンボルは、３つ以上のシンボル値のセットから選択される）を、識別子核酸配列（または識別子配列）のセットに符号化し、各固有の識別子配列は、ストリング内に対応するビットまたはシンボルを有する。識別子配列は、ストリング内のビットもしくはシンボルの位置、その値、または位置および値の両方を符号化する。本開示のシステムおよび方法を実施するための１つの方法は、図１～図１１に関連して論じるように、事前に作られたＤＮＡ成分分子（成分配列によって表される）を、画定された層に基づいて順序付けられた形で結合することによって、識別子配列によって表される各識別子核酸分子（または識別子分子）を作成することである。具体的には、異なる層内の成分配列を複数の層にわたって組合せにより組み合わせて（たとえば、１つの成分配列が層ごと

に選択される）、これらを連結（たとえば、結合）し、ストリング内の各シンボルまたはビットに１対１でマッピングされた識別子配列を形成する。

【００８４】

概して、成分核酸配列は、前記配列を含むすべての識別子に対して選択するために使用することができる１つまたは複数のプローブに結合するように構成される。たとえば、成分は、２０塩基からなる標的配列を含むことができ、プローブは、標的配列に結合するために、相補的な２０塩基のオリゴヌクレオチドを含むことができる。本開示に記載するように、各々固有のプローブに結合することが可能な成分からの識別子核酸配列の組成は、記憶されたデータに対するアクセスおよび動作に関して有益な特徴を提供する。本明細書に提示する識別子を生成する方法は、成分を含む識別子を生成するように特に構成されているが、そのような識別子核酸分子は、複数の代替の方法によって形成することもできることを理解されたい。たとえば、長さ１００塩基の核酸配列を生成するデノボ合成を使用して、各識別子が各々２０塩基からなる５つの成分を含む識別子核酸配列を作成することができる。塩基のすべての組合せが合成に対して利用可能である場合、各成分に対して最高$4^{20}$の可能な配列を得ることができる。

【００８５】

本明細書では、「シンボル」という用語は概して、デジタル情報の単位の表現を指す。デジタル情報は、シンボルストリングに分けまたは変換することができる。一例では、シンボルは、１ビットとすることができ、ビットは、「０」または「１」の値を有することができる。

【００８６】

本明細書では、「別個」または「固有」という用語は概して、グループ内の他の物体から区別可能な物体を指す。たとえば、別個または固有の核酸配列は、いかなる他の核酸配列とも同じ配列を有していない核酸配列とすることができる。別個または固有の核酸分子は、いかなる他の核酸分子とも同じ配列を有していない。別個または固有の核酸配列または分子は、別の核酸配列または分子と類似領域を共用することができる。

【００８７】

本明細書では、「成分」という用語は概して、核酸配列または核酸分子を指す。成分は、別個の核酸配列を含むことができる。成分は、１つまたは複数の他の成分と連結しまたは組み立てて、他の核酸配列または分子を生成することができる。

【００８８】

本明細書では、「層」という用語は概して、成分のグループまたはプールを指す。各層は、別個の成分のセットを含むことができ、したがって１つの層内の成分は、別の層内の成分とは異なる。１つまたは複数の層からの成分を組み立てて、１つまたは複数の識別子を生成することができる。

【００８９】

本明細書では、「識別子」という用語は概して、より大きいビットストリング内のビットストリングの位置および値を表す核酸分子または核酸配列を指す。より一般には、識別子は、シンボルストリング内のシンボルを表しまたはそれに対応する任意の物体を指すことができる。いくつかの実装形態では、識別子は、１つまたは複数の連結された成分を含むことができる。

【００９０】

本明細書では、「組合せ空間」という用語は概して、成分などの物体の開始セットから生成することができるすべての可能な別個の識別子のセット、および識別子を形成するためにそれらの物体をどのように修正するかに関する許容可能な規則セットを指す。成分を組み立てまたは連結することによって作られる識別子の組合せ空間のサイズは、成分の層の数、各層内の成分の数、および識別子を生成するために使用される特定の組立て方法に依存することができる。

【００９１】

本明細書では、「識別子ランク」という用語は概して、セット内の識別子の順序を画定

する関係を指す。

【００９２】

本明細書では、「識別子ライブラリ」という用語は概して、デジタル情報を表すシンボルストリング内のシンボルに対応する１群の識別子を指す。いくつかの実装形態では、識別子ライブラリ内の所与の識別子の不在は、特定の位置におけるシンボル値を示すことができる。１つまたは複数の識別子ライブラリは、識別子のプール、グループ、またはセットに組み合わせることができる。各識別子ライブラリは、識別子ライブラリを識別する固有のバーコードを含むことができる。

【００９３】

本明細書では、「プローブ」という用語は概して、識別子核酸分子上の標的配列に結合する剤を指す。標的配列は、成分の一部分とすることができる。プローブは、その標的配列に一致しまたはそれに相補的な配列を含むことができる。プローブは、前記標的配列を含むすべての識別子核酸分子を分離するためにさらに使用することができる。たとえば、プローブは、標的配列を含むすべての識別子核酸分子を増幅するＰＣＲ反応におけるプライマーとすることができる。別法として、プローブは、前記オリゴヌクレオチドに対応する配列を有するすべての識別子核酸分子を選択するために使用することができる親和性タグ付きオリゴヌクレオチド分子を収容することができる。

【００９４】

本明細書では、「核酸」という用語は概して、デオキシリボ核酸（ＤＮＡ）、リボ核酸（ＲＮＡ）、またはその変異体を指す。核酸は、アデノシン（Ａ）、シトシン（Ｃ）、グアニン（Ｇ）、チミン（Ｔ）、およびウラシル（Ｕ）、またはその変異体から選択された１つまたは複数のサブユニットを含みうる。ヌクレオシドは、Ａ、Ｃ、Ｇ、Ｔ、もしくはＵ、またはその変異体を含みうる。ヌクレオシドは、成長核酸鎖に組み込むことができるサブユニットを含みうる。そのようなサブユニットは、より相補的なＡ、Ｃ、Ｇ、Ｔ、もしくはＵのうちの１つに特有でありうる、またはプリン（すなわち、ＡもしくはＧ、またはその変異体）もしくはピリミジン（すなわち、Ｃ、Ｔ、もしくはＵ、またはその変異体）に相補的でありうる、Ａ、Ｃ、Ｇ、Ｔ、もしくはＵ、または他のサブユニットでありうる。いくつかの例では、核酸は１本鎖または２本鎖とすることができ、いくつかの場合、核酸は環状である。

【００９５】

本明細書では、「核酸分子」または「核酸配列」という用語は概して、様々な長さを有することができるヌクレオシドの高分子形態もしくはポリヌクレオシド、デオキシリボヌクレオシド（ＤＮＡ）もしくはリボヌクレオシド（ＲＮＡ）、またはその類似体を指す。「核酸配列」という用語は、ヌクレオシドの順序を画定するポリヌクレオシドのアルファベット表現を指し、「核酸分子」という用語は、ポリヌクレオシド自体の物理インスタンスを指す。このアルファベット表現は、中央処理装置を有するコンピュータ内のデータベースに入力することができ、核酸配列または核酸分子をシンボルまたはビットにマッピングしてデジタル情報を符号化するために使用することができる。核酸配列またはオリゴヌクレオチドは、１つまたは複数の非標準的なヌクレオシド、ヌクレオシド類似体、および／または修飾ヌクレオシドを含みうる。

【００９６】

本明細書では、「オリゴヌクレオチド」は概して、１本鎖核酸配列を指し、典型的には、アデニン（Ａ）、シトシン（Ｃ）、グアニン（Ｇ）、およびチミン（Ｔ）、またはポリヌクレオシドがＲＮＡであるときはウラシル（Ｕ）という４つのヌクレオシド塩基の特異配列から構成される。

【００９７】

修飾ヌクレオシドの例には、それだけに限定されるものではないが、ジアミノプリン、５－フルオロウラシル、５－ブロモウラシル、５－クロロウラシル、５－ヨードウラシル、ヒポキサンチン、キサンチン（xantine）、４－アセチルシトシン、５－（カルボキシヒドロキシルメチル）ウラシル、５－カルボキシメチルアミノメチル－２－チオウラジン

、５－カルボキシメチルアミノメチルウラシル、ジヒドロウラシル、ベーターＤ－ガラクトシルキュエオシン（beta-D-galactosylqueosine）、イノシン、Ｎ６－イソペンテニルアデニン、１－メチルグアニン、１－メチルイノシン、２，２－ジメチルグアニン、２－メチルアデニン、２－メチルグアニン、３－メチルシトシン、５－メチルシトシン、Ｎ６－アデニン、７－メチルグアニン、５－メチルアミノメチルウラシル、５－メトキシアミノメチル－２－チオウラシル、ベーターＤ－メノシルキュエオシン（beta-D-mannosylqueosine）、５’－メトキシカルボキシメチルウラシル、５－メトキシウラシル、２－メチルチオ－Ｄ４６－イソペンテニルアデニン、ウラシル－５－オキシ酢酸（ｖ）、ワイブトキソシン、シュードウラシル、キュエオシン（queosine）、２－チオシトシン、５－メチル－２－チオウラシル、２－チオウラシル、４－チオウラシル、５－メチルウラシル、ウラシル－５－オキシ酢酸メチルエステル、ウラシル－５－オキシ酢酸（ｖ）、５－メチル－２－チオウラシル、３－（３－アミノ－３－Ｎ－２－カルボキシプロピル）ウラシル、（ａｃｐ３）ｗ、２，６－ジアミノプリンなどが含まれる。核酸分子はまた、塩基部分（たとえば、典型的に相補的ヌクレオシドとの水素結合を形成するために利用可能な１つもしくは複数の原子、および／または典型的に相補的ヌクレオシドとの水素結合を形成することが可能でない１つもしくは複数の原子）、糖部分、またはリン酸主鎖で修飾することができる。核酸分子はまた、Ｎ－ヒドロキシスクシンイミドエステル（ＮＨＳ）などのアミン反応部分の共有結合を可能にするために、アミノアリル－ｄＵＴＰ（ａａ－ｄＵＴＰ）およびアミノヘキシルアクリルアミド－ｄＣＴＰ（ａｈａ－ｄＣＴＰ）などのアミン修飾基を含有することができる。

【００９８】

本明細書では、「プライマー」という用語は概して、ポリメラーゼ連鎖反応（ＰＣＲ）などの核酸合成開始点として働く核酸の鎖を指す。一例では、ＤＮＡサンプルの複製中、複製に触媒作用を及ぼす酵素は、ＤＮＡサンプルに取り付けられたプライマーの３’末端で複製を開始し、逆鎖をコピーする。

【００９９】

本明細書では、「ポリメラーゼ」または「ポリメラーゼ酵素」という用語は概して、ポリメラーゼ反応に触媒作用を及ぼすことが可能な酵素を指す。ポリメラーゼの例には、限定ではないが、核酸ポリメラーゼが含まれる。ポリメラーゼは、自然に発生または合成することができる。例示的なポリメラーゼは、Φ２９のポリメラーゼまたはその誘導体である。いくつかの場合、ポリメラーゼとともに、またはポリメラーゼの代替として、新しい核酸配列を構築するために、転写酵素またはリガーゼ（すなわち、結合の形成に触媒作用を及ぼす酵素）が使用される。ポリメラーゼの例には、ＤＮＡポリメラーゼ、ＲＮＡポリメラーゼ、熱安定性ポリメラーゼ、野生型ポリメラーゼ、修飾ポリメラーゼ、Ｅ．ｃｏｌｉ　ＤＮＡポリメラーゼＩ、Ｔ７ＤＮＡポリメラーゼ、バクテリオファージＴ４　ＤＮＡポリメラーゼΦ２９（ｐｈｉ２９）ＤＮＡポリメラーゼ、Ｔａｑポリメラーゼ、Ｔｔｈポリメラーゼ、Ｔｌｉポリメラーゼ、Ｐｆｕポリメラーゼ、Ｐｗｏポリメラーゼ、ＶＥＮＴポリメラーゼ、ＤＥＥＰＶＥＮＴポリメラーゼ、Ｅｘ－Ｔａｑポリメラーゼ、ＬＡ－Ｔａｗポリメラーゼ、Ｓｓｏポリメラーゼ、Ｐｏｃポリメラーゼ、Ｐａｂポリメラーゼ、Ｍｔｈポリメラーゼ、ＥＳ４ポリメラーゼ、Ｔｒｕポリメラーゼ、Ｔａｃポリメラーゼ、Ｔｎｅポリメラーゼ、Ｔｍａポリメラーゼ、Ｔｃａポリメラーゼ、Ｔｉｈポリメラーゼ、Ｔｆｉポリメラーゼ、プラチナＴａｑポリメラーゼ、Ｔｂｒポリメラーゼ、Ｔｆｌポリメラーゼ、Ｐｆｕｔｕｂｏポリメラーゼ、Ｐｙｒｏｂｅｓｔポリメラーゼ、ＫＯＤポリメラーゼ、Ｂｓｔポリメラーゼ、Ｓａｃポリメラーゼ、３’～５’エキソヌクレアーゼ活性を有するＫｌｅｎｏｗ断片ポリメラーゼ、ならびにそれらの変異体、修飾産物、および誘導体が含まれる。

【０１００】

２進コードの形式のコンピュータデータなどのデジタル情報は、シンボル配列またはストリングを含むことができる。２進コードは、たとえば２つの２進シンボル、典型的にはビットと呼ばれる０および１を有する２進法を使用して、テキストまたはコンピュータプ

ロセッサ命令を符号化しまたは表すことができる。デジタル情報は、非２進シンボルの配列を含むことができる非２進コードの形式で表すことができる。符号化された各シンボルは、固有のビットストリング（または「バイト」）に再び割り当てることができ、固有のビットストリングまたはバイトは、バイトまたはバイトストリームのストリングに配置することができる。所与のビットに対するビット値は、２つのシンボル（たとえば、０または１）のうちの１つとすることができる。バイトは、Ｎビットのストリングを含むことができ、合計$２^{N}$の固有のバイト値を有することができる。たとえば、８ビットを含むバイトは、合計$２^{8}$または２５６の可能な固有のバイト値を作り出すことができ、２５６バイトの各々は、バイトによって符号化することができる２５６個の可能な別個のシンボル、文字、または命令のうちの１つに対応することができる。生データ（たとえば、テキストファイルおよびコンピュータ命令）は、バイトまたはバイトストリームのストリングとして表すことができる。生データを含むＺｉｐファイルまたは圧縮データファイルもまた、バイトストリームに記憶することができ、これらのファイルは、圧縮された形式でバイトストリームとして記憶することができ、次いで生データに解凍してから、コンピュータによって読み取ることができる。

【０１０１】

「索引」および「位置」という用語は、本開示で区別なく使用され、どちらの用語も、リストまたはストリングなどの順序付けられた群の特有の要素または実体を指すために使用されることを理解されたい。たとえば、索引または位置を使用して、アレイ、ベクトル、ストリング、またはデータ構造内の要素を指定することができる。索引／位置の表記は、付番方式を使用して、各エントリ／実体に公称番号を割り当てる。本開示の例は、当技術分野では公知の第１の索引／位置０を、０に基づく付番として使用することが多い。アレイ／ストリングの第１の位置（第０の位置とも呼ばれる）は、特有の位置を含む算出の目的で、０によって示される。１組の長さｎは、０、１、．．．、ｎ－１の付番方式を有するはずである。本明細書に記載するシステムおよび方法では、他の付番方式を使用することもできることを理解されたい。たとえば、付番方式は、１組の長さｎに対して、１で開始し、ｎまで継続することができる。

【０１０２】

本開示は、本出願の図に関連する方法について説明する。これらの方法は、算出ステップを含み、ＤＮＡで実行されるように構成されることを理解されたい。本開示の方法およびシステムを使用して、コンピュータデータまたは情報を複数の識別子で符号化することができ、識別子の各々は、元の情報の１つまたは複数のビットを表すことができる。いくつかの例では、本開示の方法およびシステムは、識別子を使用してデータまたは情報を符号化し、識別子の各々は、元の情報の２つのビットを表す。

【０１０３】

核酸配列への情報の符号化および書込み

【０１０４】

以下の説明では、図１～図１０に関連して、核酸分子でデータを符号化する様々なシステムおよび方法の概略を提供し、符号化された核酸分子を印刷および記憶するように構成された様々な書込みおよび格納システムについて説明する。

【０１０５】

図１は、例示的な実装形態による核酸配列への情報の符号化、核酸配列への情報の書込み、核酸配列に書き込まれた情報の読取り、および読み取った情報の復号のための概略的なプロセスを示す。デジタル情報またはデータは、１つまたは複数のシンボルストリングとして表される。一例では、シンボルはビットであり、各ビットは「０」または「１」の値を有する。各シンボルは、そのシンボルを表すオブジェクト（たとえば、識別子）にマッピングまたは符号化することができ、したがって各シンボルは、別個の識別子によって表される。別個の識別子は、成分核酸配列（本明細書では、成分と呼ぶことができる）から構築された特異核酸配列を有する１つまたは複数の核酸分子とすることができる。デジタル情報を核酸配列に書き込むために、プロセスは、デジタル情報の各シンボルに対応す

る識別子を物理的に構築することによって物理的に生成することができる「識別子ライブラリ」を生成する。識別子ライブラリからデジタル情報を読み取るために、プロセスは、識別子をシークエンシングおよび識別することによって、識別子ライブラリ内の識別子のすべてまたはサブセットにアクセスする。次いで、識別された識別子を、対応するシンボルに関連付けて、元のデジタルデータを復号する。概して、本手法は、一度にデジタル情報のすべてまたは任意の部分にアクセスすることを可能にする。

【０１０６】

一例では、図１の手法を使用した情報の符号化および読取りのための方法は、ビットストリームを受け取ることと、識別子ランクまたは核酸索引を使用して、ビットストリーム内の各１ビット（「１」のビット値を有するビット）を別個の核酸識別子にマッピングすることとを含む。次いで、プロセスは、１のビット値に対応する識別子のコピーを含む核酸分子のプールとして、識別子ライブラリを構築する（ビット値０の場合は識別子を除外する）。言い換えれば、識別子ライブラリは、ビットストリーム内の特有の位置またはランクにおいて１ビットを表す各識別子の複数のコピーを含み、各識別子は、ビットストリーム内の特有の位置またはランクを表す特異配列を共用するプール内の識別子分子の複数のインスタンスに関連付けられており、ビットストリーム内の０ビットを表す識別子は、プールから除外される。核酸分子のプールからデジタルデータを読み取るために、分子生物学方法（たとえば、シークエンシング、ハイブリダイゼーション、ＰＣＲなど）を使用して、どの識別子が識別子ライブラリ内に表されているかを判定することができる。識別子ライブラリ内に存在する識別子の場合、対応する識別子ランク（したがって、ビットストリーム内のビット位置）が判定され、ビット値「１」がその場所に割り当てられる。識別子ライブラリに存在しないあらゆる識別子に対しては、対応する識別子ランク（したがって、ビットストリーム内のビット位置）が判定され、ビット値「０」がその場所に割り当てられる。このようにして、核酸分子のプールを復号して、元の符号化されたビットストリームを判定することができる。

【０１０７】

上述した手法は、Ｎ個の別個のビットのストリングを符号化することを伴い、等しい数Ｎの固有の核酸配列を可能な識別子として用いる。情報の符号化に対するこの手法は、記憶すべき情報の新しい各項目（Ｎビットの別のストリング）に対して、識別子（たとえば、核酸分子）のデノボ合成を使用することができる。他の事例では、記憶すべき情報の新しい各項目に対して識別子（数がＮに等しいまたはそれよりも小さい）を新しく合成するコストを低減させて、すべての可能なＮ個の識別子の１回限りのデノボ合成および後の保守にすることができる。このようにして、情報の新しい項目（たとえば、長さＮまたはそれよりも小さいビットストリング）を符号化することは、事前に合成（または事前に製作）された識別子を機械的に選択してともに混合し、識別子ライブラリを形成することを伴う。他の事例では、核酸配列の数（Ｎより小さい、いくつかの場合はＮよりはるかに小さい）を合成および維持し、次いで酵素反応によってこれらの配列を修飾して、記憶すべき情報の新しい各項目に対して最大Ｎ個の識別子を生成することによって、（１）記憶すべき情報の新しい各項目に対する最大Ｎ個の識別子のデノボ合成、もしくは（２）記憶すべき情報の新しい各項目に対するＮ個の可能な識別子からの維持および選択、またはその両方のコストを低減させることができる。たとえば、合成および維持される核酸配列は、成分などのＮ個の識別子を構成する特有の部分に対応することができる。

【０１０８】

識別子は、読取り、書込み、アクセス、コピー、および削除演算を容易にするように合理的に設計および選択することができる。識別子は、書込みエラー、突然変異、劣化、および読取りエラーを最小にするように設計および選択することができる。

【０１０９】

特有の例として、利用可能な識別子配列のセットは、１５の層を含むことができ、そのうち１４層は各々、６つの固有のＤＮＡ成分配列を収容する。第１５の層は、２８個のＤＮＡ成分配列（６つではない）を含む多重化層とすることができ、これらのＤＮＡ成分配

列も組み込まれる。したがって、各識別子は、識別子核酸分子の全長にわたって、１５個の成分（各層に１成分）を収容することができる。書込みプロセス中、成分分子を反応コンパートメント内にともに組み立てて、識別子分子を形成する。いくつかの実装形態では、「多重化層」のみからの複数の成分が、同じ反応コンパートメント内へ組み合わせられる。

【０１１０】

別の例として、８６，４００秒（２４時間）で１テラバイトを書き込むために、約８×$10^{11}$個の識別子分子を組み立てること（１つの識別子当たり１０ビットの情報が符号化されると仮定する）、または約５×$10^{10}$の液滴反応コンパートメントが必要となりうる。各反応は、２８個の識別子の可能なセットから、１４個の識別子を組み立てることができる。１４個の成分（１４層の各々から１つ、各層は６つの可能な成分を有する）は、識別子の「塩基」を指定して組み立てる。多重化層からの２８個の可能な成分のうち残りの１４個の成分は、どの１４個の識別子（２８の可能性から）が組み立てられるかを指定する。したがって、各反応コンパートメントは、２８個のＤＮＡ成分と、リガーゼまたは他の反応混合物とを必要とすることができる。

【０１１１】

本明細書に記載する方法は、後述するように、書込みシステムを使用して実施することができる。書込みシステムは、参照により本明細書に組み込まれている、２０１９年５月１６日出願の"Printer-Finisher System for Data Storage in DNA"という名称の米国特許出願第１６／４１４，７５２号に記載されているものなどのプリンタフィニッシャシステムとすることができる。ライタシステムは、基質上の個別の場所（たとえば、反応コンパートメント）にＤＮＡ成分を分注し、結合マスタ混合物を分注し、結合反応に最適の条件を提供し、ライブラリを含むＤＮＡ識別子のすべてをプールすることができる。

【０１１２】

本明細書に記載する書込みシステムは、識別子を構築するために、結合反応の高処理量の並列化された印刷を実行することが可能である。反応は、ローラの上を動く可撓シート（ウェビングまたは基質とも呼ばれる）上へ印刷されるピコリットル（ｐＬ）規模の液滴で実施することができる。上記で参照した出願に記載されているように、書込みシステムは、好適な既製の印刷ヘッド、ドライバ、および機械インフラストラクチャを使用するデジタルインクジェット印刷およびウェブハンドリングなどの技術を組み込むことができる。いくつかの実装形態では、本明細書に記載するシステムおよび方法は、記憶容量および書込み処理量を実現するために、ウェブ速度、印刷ヘッド分注速度、液滴サイズ、および結合の化学的性質などの要因の最適化を含む。この目的で、潜在的な化学的性質およびハードウェアエラーに対するデータ許容度を確保するために、本明細書に記載するシステムおよび方法は、ＤＮＡ成分配列をどのように層に区画化するか、および各印刷反応においてどれだけの識別子分子を構築するかに関する仕様を含めて、データの符号化および印刷命令の展開のための構成を含む。たとえば、そのような構成は、書込みシステムと通信してその性能を追跡するコンピュータシステムを含むことができる。

【０１１３】

図２Ａおよび図２Ｂは、例示的な実装形態によるデジタルデータをオブジェクトまたは識別子（たとえば、核酸分子）に符号化する「データアットアドレス」と呼ばれる例示的な方法を概略的に示す。図２Ａは、ビットストリームを識別子ライブラリに符号化することを示し、個々の識別子は、識別子ランクを指定する単一の成分を、バイト値を指定する単一の成分に連結しまたは組み立てることによって構築される。概して、データアットアドレス方法は、２つのオブジェクトを含むことによって情報をモジュール式に符号化する識別子を使用し、１つのオブジェクトは、バイト値を識別する「バイト値オブジェクト」（または「データオブジェクト」）であり、１つのオブジェクトは、識別子ランク（または元のビットストリーム内のバイトの相対位置）を識別する「ランクオブジェクト」（または「アドレスオブジェクト」）である。図２Ｂは、データアットアドレス方法の一例を示し、各ランクオブジェクトは、成分のセットから組合せにより構築することができ、各

バイト値オブジェクトは、成分のセットから組合せにより構築することができる。ランクおよびバイト値オブジェクトのそのような組合せ構造により、オブジェクトが単一の成分のみから作られた場合（たとえば、図２Ａ）より、多くの情報を識別子に書き込むことが可能になる。

【０１１４】

図３Ａおよび図３Ｂは、例示的な実装形態によるデジタル情報をオブジェクトまたは識別子（たとえば、核酸配列）に符号化する別の例示的な方法を概略的に示す。図３Ａは、ビットストリームを識別子ライブラリに符号化することを示し、識別子は、ビットストリーム内の位置に対応する識別子ランクを指定する単一の成分から構築される。特定のランク（またはアドレス）における識別子の存在は、ビット値「１」を指定し、特定のランク（またはアドレス）における識別子の不在は、ビット値「０」を指定する。このタイプの符号化は、ランク（元のビットストリーム内のビットの相対位置）のみを符号化する識別子を使用することができ、識別子ライブラリ内のそれらの識別子の有無を使用して、それぞれビット値「１」または「０」を符号化することができる。情報の読取りおよび復号は、識別子ライブラリ内に存在する識別子を識別することと、対応するランクにビット値「１」を割り当てることと、そうでない場合はビット値「０」を割り当てることとを含むことができる。この例では、識別子の存在が１ビットを符号化し、識別子の不在が０ビットを符号化するが、本開示の範囲から逸脱することなく、識別子の存在が０ビットを符号化し、識別子の不在が１ビットを符号化することもできることが理解されよう。

【０１１５】

図３Ｂは、図３Ａに類似しているが、図３Ｂの例示的な符号化方法では、各識別子は、可能な各組合せ構造がランクを指定するように、成分のセットから組合せにより構築される。そのような組合せ構造により、識別子が単一の成分のみから作られた場合（たとえば、図３Ａ）より、多くの情報を識別子に書き込むことが可能になる。たとえば、図３Ｂに描くように、５つの別個の成分からなる成分セットを使用して、長さＮ＝１０のビットストリングに対応する１０個のアドレスが表される。５つの別個の成分は、１０個の別個の識別子を生成するように、組合せによって組み立てられ、各識別子は、５つの成分のうちの２つを含む。１０個の別個の識別子は各々、ビットストリーム内のビットの位置に対応するランク（またはアドレス）を有する。識別子ライブラリは、長さ１０のビットストリーム内で、ビット値「１」の位置に対応するそれらの可能な１０個の識別子のサブセットを含むことができ、ビット値「０」の位置に対応するそれらの可能な１０個の識別子のサブセットを除外することができる。

【０１１６】

図４は、図３Ａおよび図３Ｂに示す符号化方法を使用してビット単位の所与の元のサイズの情報（Ｄ、等高線）を記憶するように物理的に構築される可能な識別子の組合せ空間（Ｃ、ｘ軸）と識別子の平均数（ｋ、ｙ軸）との間の関係の対数空間における例示的な実装形態による等高線図を示す。このグラフは、サイズＤの元の情報が、Ｃ個のビットのストリングに再コード化され（ＣはＤより大きくすることができる）、指定数ｋのビットがビット値「１」を有すると仮定する。さらに、グラフは、情報から核酸への符号化が、再コード化されたビットストリングで実行され、ビット値が「１」になる位置に対する識別子が構築され、ビット値が「０」になる位置に対する識別子が構築されないと仮定する。この仮定に従って、可能な識別子の組合せ空間は、再コード化されたビットストリング内のすべての位置を識別するためにサイズＣを有し、サイズＤのビットストリングを符号化するために使用される識別子の数は、Ｄ＝ｌｏｇ２（Ｃｃｈｏｏｓｅｋ）になるようになっており、ここでＣｃｈｏｏｓｅｋは、Ｃ個の可能性からｋ個の順序付けられていない結果を選ぶ方法の数である。したがって、可能な識別子の組合せ空間が情報の所与の項目のサイズ（ビット単位）を超えて増大すると、所与の情報を記憶するために必要な物理的に構築された識別子の数が減少する。

【０１１７】

図５は、例示的な実装形態によって核酸配列に情報を書き込む概略的な方法を示す。例

示的な実装形態によれば、情報は、核酸分子に書き込まれる前に、シンボルストリングに変換され、複数の識別子配列に符号化される。一例では、核酸分子に情報を書き込むことは、プール内に組み合わせるための識別子分子を作り出すように様々な化学反応を設定することを含む。具体的には、入力（たとえば、核酸、成分、鋳型、酵素、または化学試薬など）をコンパートメント（本明細書ではコンテナとも呼ばれる、たとえばウェル、チューブ、表面上の位置、マイクロ流体デバイス内のチャンバ、または乳剤内の液滴など）内へ堆積させることによって、反応が設定される。反応は、一度に単一のコンパートメント内に設定することができ、または並列処理のために複数のコンパートメント内に設定することができる。反応は、プログラムされた温度のインキュベーションもしくは循環などの特有のプロセスステップを含むことができ、選択的もしくは普遍的に除去（たとえば、削除）することができ、その結果得られる識別子分子を１つのプール内に収集するように選択的もしくは普遍的に中断、統合、および精製することができ、またはこれらの任意の好適な組合せとすることができる。複数の識別子ライブラリからの識別子は、同じプール内に収集することができ、または異なる識別子ライブラリの各々を、別個の個々のプール内に収集することができる。いくつかの例では、個々の識別子は、その識別子が属する対応する識別子ライブラリを識別するために、固有のバーコードまたはタグを含む。いくつかの例では、そのバーコードは、符号化された情報を表すメタデータを含む。符号化された情報自体を表す識別子分子に加えて、補足の核酸または追加の識別子を識別子ライブラリとともに識別子プール内に含むこともできる。たとえば、補足の核酸または追加の識別子は、符号化された情報に対するメタデータを表すことができ、または符号化された情報を不明瞭にしもしくは隠す働きをすることもできる。

【０１１８】

識別子ランク（たとえば、核酸索引）は、識別子の順序付けに基づいている。この方法は、すべての識別子および対応するランクを有するルックアップテーブルを含むことができる。この方法はまた、識別子を構成するすべての成分のランクおよびそれらの成分の組合せを含むあらゆる識別子の順序付けを判定するための関数を有するルックアップテーブルを含むことができる。そのような方法を、辞書的順序付けと呼ぶことができ、辞書内の単語がアルファベット順で順序付けられる方法に類似したものとすることができる。たとえば、組合せ空間内の各識別子は、固定数のＮ個の成分を含むことができ、各成分は、Ｎ個の層のセット内の別個の層に由来し、前記層内の複数の可能な成分のセットのうちの１つである。各成分は、座標（ｊ，Ｘｊ）によって指定することができ、ここでｊは層のラベルであり、Ｘｊは層内の成分のラベルである。Ｎ個の層を伴う前記方式の場合、ｊは集合｛１，２，．．．，Ｎ｝の要素であり、Ｘｊは、集合｛１，２，．．．，Ｍｊ｝の要素であり、ここでＭｊは層ｊ内の成分の数である。本発明者らは、これらの層に対する論理順序を画定することができる。本発明者らはまた、各層内の各成分に対する論理順序を画定することができる。本発明者らは、このラベル付けを使用して、関数またはアルゴリズムによって、組合せ空間内のすべての可能な識別子に対する論理順序付けを画定することができる。たとえば、本発明者らは、まず層１内の成分の順序に従って、次いで後に層２内の成分の順序に従って、識別子を分類することができ、以下同様である。

【０１１９】

データアットアドレス符号化方法では、識別子ランク（識別子のランクオブジェクトによって符号化される）を使用して、ビットストリーム内のバイト（識別子のバイト値オブジェクトによって符号化される）の位置を判定することができる。代替方法では、存在する識別子に対する識別子ランク（識別子自体の全体によって符号化される）を使用して、ビットストリーム内のビット値「１」の位置を判定することができる。特有のランクを判定しまたは１つまたは複数の識別子のランクを使用する様々な方法について説明するシステムおよび方法は、図２３～図２９に関連して説明する。

【０１２０】

鍵が、サンプル内の識別子（たとえば、核酸分子）の固有のサブセットに、別個のバイトまたは情報部分を割り当てることができる。たとえば、簡単な形式では、鍵は、ビット

の位置を指定する固有の核酸配列に、バイト内の各ビットを割り当てることができ、次いでサンプル内のその核酸配列の有無が、それぞれビット値１または０を指定することができる。本開示の範囲を逸脱することなく、他のタイプの鍵を使用することもできる。符号化された情報を核酸サンプルから読み取ることは、シークエンシング、ハイブリダイゼーション、またはＰＣＲを含む任意の数の分子生物学技法を含むことができる。いくつかの実装形態では、符号化されたデータセットを読み取ることは、データセットの一部分を再構築すること、または符号化されたデータセット全体を各核酸サンプルから再構築することを含むことができる。配列が読み取られるとき、固有の核酸配列の有無とともに核酸索引が判定され、核酸サンプルをビットストリームに復号することができる（たとえば、各ストリングは、複数ビット、１バイト、複数バイト、またはバイトストリングである）。

【０１２１】

いくつかの実装形態では、成分核酸配列を組合せにより組み立てることによって、識別子が構築される。たとえば、画定された分子のグループ（たとえば、組合せ空間）から核酸分子（たとえば、識別子）のセットを得ることによって、情報を符号化することができる。画定された分子のグループの可能な各識別子は、層に分けることができる事前製作された成分のセットからの核酸配列（たとえば、成分）のアセンブリとすることができる。個々の各識別子は、すべての層からの１つの成分を固定の順序で連結することによって構築することができる。たとえば、Ｍ個の層が存在し、各層がｎ個の成分を有することができる場合、最大Ｃ＝$n^M$個の固有の識別子を構築することができ、最大$2^C$の異なる項目の情報またはＣビットを符号化および記憶することができる。たとえば、１メガビットの情報の記憶は、$1\times10^6$個の別個の識別子またはサイズ$C=1\times10^6$の組合せ空間を使用することができる。この例では、識別子は、異なる方法で組織化された様々な成分から組み立てることができる。アセンブリは、Ｍ＝２個の事前製作された層から作ることができ、各層は$n=1\times10^3$個の成分を収容する。別法として、アセンブリは、Ｍ＝３個の層から作ることができ、各層は$n=1\times10^2$個の成分を収容する。いくつかの実装形態では、アセンブリは、Ｍ＝２、Ｍ＝３、Ｍ＝４、Ｍ＝５、またはそれよりも多い層から作ることができる。この例が示すように、より多数の層を使用して同じ量の情報を符号化することで、成分の総数をより小さくすることを可能にすることができる。全体でより少数の成分を使用することは、書込みコストの点から有利となりうる。

【０１２２】

一例では、２セットの固有の核酸配列または層ＸおよびＹが存在し、各核酸配列は、それぞれｘおよびｙの成分（たとえば、核酸配列）を有する。Ｘからの各核酸配列を、Ｙからの各核酸配列に組み立てることができる。２つのセット内で維持される核酸配列の総数は、ｘおよびｙの和であるが、生成することができる核酸分子、したがって可能な識別子の総数は、ｘおよびｙの積である。任意の順序でＸからの配列をＹの配列に組み立てることができる場合、さらに多くの核酸配列（たとえば、識別子）を生成することができる。たとえば、組立て順序がプログラム可能である場合、生成される核酸配列（たとえば、識別子）の数を、ｘおよびｙの積の２倍にすることができる。生成することができるすべての可能な核酸配列のこのセットを、ＸＹと呼ぶことができる。ＸＹ内の固有の核酸配列の組み立てられたユニットの順序は、別個の５’および３’末端を有する核酸を使用して制御することができ、配列の別個の５’および３’末端に対して、制限消化、結合、ポリメラーゼ連鎖反応（ＰＣＲ）、およびシークエンシングを行うことができる。したがって、すべてのユニットおよび必要な試薬を反応コンパートメント内に同時に堆積させることができ、組み立てられたユニットの順位が末端の設計によって制御されるため、各ユニットの別個の５’および３’末端により、ユニット（たとえば、成分）が所望の固有の核酸分子に自己組織化することが可能になる。そのような手法は、組立て積の組合せおよび順序で情報を符号化することによって、Ｎ個の別個のビットを符号化するために使用される核酸配列（たとえば、成分）の総数を低減させることができる。たとえば、１００ビットの情報を符号化するために、１０個の別個の核酸分子（たとえば、成分）の２つの層を固定の順序で組み立てて、$10^2$または１００個の別個の核酸分子（たとえば、識別子）を作り出す

ことができ、または５個の別個の核酸分子（たとえば、成分）の１つの層と、１０個の別個の核酸分子（たとえば、成分）の別の層とを任意の順序で組み立てて、１００個の別個の核酸分子（たとえば、識別子）を作り出すことができる。

【０１２３】

各層内の核酸配列（たとえば、成分）は、中央の固有（または別個）の配列またはバーコード、一方の末端の共通のハイブリダイゼーション領域、および他方の末端の別の共通のハイブリダイゼーション領域を含むことができる。バーコードは、層内のすべての配列を固有に識別するのに十分な数のヌクレオシドを収容することができる。たとえば、典型的には、バーコード内の各塩基位置に対して、４つの可能なヌクレオシドが存在する。したがって、３つの塩基バーコードが、$4^3=64$個の核酸配列を一意に識別することができる。バーコードは、ランダムに生成されるように設計することができる。別法として、バーコードは、識別子またはシークエンシングの構造の化学的性質に複雑さを生じさせる可能性のある配列を回避するように設計することができる。加えて、バーコードは、各バーコードが他のバーコードから最小のハミング距離を有することができるように設計することができ、それによって塩基分解能の突然変異または読取りエラーがバーコードの適切な識別に干渉しうる可能性を減少させることができる。また、バーコード領域は、ＰＣＲに対するプライマー、ＣＲＩＳＰＲ－ＣａｓガイドＲＮＡ、または親和性タグ付きオリゴヌクレオチド（たとえば、ビオチン化オリゴヌクレオチド）などのプローブに結合するように設計することができる。

【０１２４】

核酸配列（たとえば、成分）の一方の末端のハイブリダイゼーション領域は、各層内で異なることもあるが、ハイブリダイゼーション領域は、層内の各部材に対して同じにすることができる。隣接する層は、それらの成分に相補的なハイブリダイゼーション領域を有するものであり、したがって互いに相互作用することが可能である。たとえば、相補的なハイブリダイゼーション領域を有することができるため、層Ｘからのあらゆる成分は、層Ｙからのあらゆる成分に取り付けることが可能である。反対の末端のハイブリダイゼーション領域は、第１の末端のハイブリダイゼーション領域と同じ目的を担うことができる。たとえば、層Ｙからのあらゆる成分は、一方の末端で層Ｘのあらゆる成分に取り付けることができ、反対の末端で層Ｚのあらゆる成分に取り付けることができる。したがって、複数の識別子を形成するためのすべての成分および必要な試薬を反応コンパートメント内に同時に堆積させることができ、組み立てられた成分の順序がハイブリダイゼーション領域の設計によって制御されるため、各成分上のハイブリダイゼーション領域により、成分が所望の固有の識別子分子に自己組織化することが可能になる。

【０１２５】

図６Ａおよび図６Ｂは、例示的な実装形態による各層からの別個の成分（たとえば、核酸配列）を固定の順序で組合せにより組み立てることによって、識別子（たとえば、核酸分子）を構築するための「積方式」と呼ばれる例示的な方法を示す。図６Ａは、積方式を使用して構築された識別子のアーキテクチャを示す。識別子は、各層からの単一の成分を固定の順序で組み合わせることによって構築することができる。各々Ｎ個の成分を有するＭ個の層の場合、$N^M$個の可能な識別子が存在する。図６Ｂは、積方式を使用して構築することができる識別子の組合せ空間の一例を示す。一例では、組合せ空間は、３つの層から生成することができ、各層は、３つの別個の成分を含む。これらの成分は、各層からの１つの成分を固定の順序で組み合わせることができるように組み合わせることができる。この組立て方法に対する組合せ空間全体は、２７個の可能な識別子を含むことができる。

【０１２６】

図７～図１０は、積方式（図６参照）を実施するための化学的方法を示す。図７～図１０に描く方法は、２つまたはそれよりも多い別個の成分を固定の順序で組み立てるための任意の他の方法とともに、たとえば識別子ライブラリ内にいずれか１つまたは複数の識別子を作り出すために使用することができる。これらの方法は、全体として参照により組み込まれている、２０１７年１２月２１日出願の"NUCLEIC ACID-BASED DATA STOR

AGE"という名称の米国特許第１０，６５０，３１２号に記載されている。本明細書に開示する方法またはシステム中の任意の時点で、図７～図１０に記載する実装方法のいずれかを使用して、識別子を構築することができる。いくつかの事例では、可能な識別子の組合せ空間のすべてのまたは一部分は、デジタル情報が符号化されまたは書き込まれる前に構築することができ、このとき書込みプロセスは、既存のセットから識別子（情報を符号化する）を機械的に選択およびプールすることを伴うことができる。他の事例では、識別子は、データ符号化または書込みプロセスの１つまたは複数のステップが行われた後（すなわち、情報が書き込まれるとき）に構築することができる。

【０１２７】

酵素反応を使用して、異なる層またはセットからの成分を組み立てることができる。組立ては、各層の成分（たとえば、核酸配列）が隣接する層の成分に対して特有のハイブリダイゼーションまたは取付け領域を有するため、１ポット反応で行うことができる。たとえば、層Ｘからの核酸配列（たとえば、成分）Ｘ１、層Ｙからの核酸配列Ｙ１、および層Ｚからの核酸配列Ｚ１が、組み立てられた核酸分子（たとえば、識別子）Ｘ１Ｙ１Ｚ１を形成することができる。加えて、各層から複数の核酸配列を含むことによって、複数の核酸分子（たとえば、識別子）を１つの反応で組み立てることができる。１つの反応は、成分の識別子への自己組織化を伴うことができる。

【０１２８】

例示的な実装形態によれば、図７に示すように、オーバーラップ伸長ポリメラーゼ連鎖反応（ＯＥＰＣＲ）を使用して積方式に従って識別子を構築することができる。各層内の各成分は、配列末端に共通のハイブリダイゼーション領域を有する２本鎖または１本鎖（図示）の核酸配列を含むことができ、共通のハイブリダイゼーション領域は、隣接する層からの成分の配列末端の共通のハイブリダイゼーション領域に相同および／または相補的とすることができる。したがって、複数の識別子を形成するためのすべての成分および必要な試薬を反応コンパートメント内に同時に堆積させることができ、組み立てられた成分の順序がハイブリダイゼーション領域の設計によって制御されるため、各成分上のハイブリダイゼーション領域により、成分が所望の固有の識別子分子に自己組織化することが可能になる。

【０１２９】

例示的な実装形態によれば、図８に示すように、付着末端結合を使用して積方式に従って識別子を組み立てることができる。各々１本鎖３’オーバーハングを有する２本鎖成分（たとえば、２本鎖ＤＮＡ（ｄｓＤＮＡ））を含む３つの層を使用して、別個の識別子を組み立てることができる。付着末端結合のための付着末端は、制限エンドヌクレアーゼで各層の成分を処理することによって生成することができる。いくつかの実装形態では、１つの「親」セットの成分から、複数の層の成分を生成することができる。

【０１３０】

例示的な実装形態によれば、図９に示すように、部位特異的組換えを使用して積方式に従って識別子を組み立てることができる。３つの異なる層からの成分を組み立てることによって、識別子を構築することができる。層Ｘ（または層１）内の成分は、分子の一方の側にａｔｔＢｘリコンビナーゼ部位を有する２本鎖分子を含むことができ、層Ｙ（または層２）からの成分は、一方の側のａｔｔＰ$_x$リコンビナーゼ部位および他方の側のａｔｔＢ$_y$リコンビナーゼ部位を有する２本鎖分子を含むことができ、層Ｚ（または層３）内の成分は、分子の一方の側にａｔｔＰ$_y$リコンビナーゼ部位を含むことができる。添字によって示される１対の中のａｔｔＢおよびａｔｔＰ部位は、対応するリコンビナーゼ酵素の存在下で組み換えることが可能である。各層からの１つの成分は、層Ｘからの１つの成分が層Ｙからの１つの成分に関連し、層Ｙからの１つの成分が層Ｚからの１つの成分に関連するように組み合わせることができる。したがって、複数の識別子を形成するためのすべての成分および必要な試薬を反応コンパートメント内に同時に堆積させることができ、組み立てられた成分の順序がリコンビナーゼ部位の設計によって制御されるため、各成分上のリコンビナーゼ部位により、成分が所望の固有の識別子分子に自己組織化することが可

能になる。

【０１３１】

例示的な実装形態によれば、図１０Ａに示すように、鋳型指向結合（ＴＤＬ）を使用して積方式に従って識別子を構築することができる。鋳型指向結合は、「鋳型」または「ステープル」と呼ばれる１本鎖核酸配列を利用して、識別子を形成するための成分の順序付けられた結合を容易にする。鋳型は、隣接する層からの成分に同時にハイブリッド化し、リガーゼが成分を結合する間に、成分を互いに（３’末端を５’末端に）隣接して保持する。したがって、複数の識別子を形成するためのすべての成分および必要な試薬を反応コンパートメント内に同時に堆積させることができ、組み立てられた成分の順序が鋳型の設計によって制御されるため、各成分上のハイブリダイゼーション領域により、成分が所望の固有の識別子分子に自己組織化することが可能になる。

【０１３２】

図１０Ｂは、例示的な実装形態による各々６層ＴＤＬによって組み立てられた２５６個の別個の核酸配列のコピー数（存在度）のヒストグラムを示す。縁部の層（第１および最後の層）は各々、１つの成分を有し、内部の層（残り４つの層）の各々は、４つの成分を有した。各縁部層成分は、１０塩基のハイブリダイゼーション領域を含む２８塩基であった。各内部層成分は、５’末端の１０塩基の共通のハイブリダイゼーション領域、１０塩基の可変（バーコード）領域、および３’末端の１０塩基の共通のハイブリダイゼーション領域を含む３０塩基であった。３つの鋳型鎖の各々は、長さ２０塩基であった。すべての２５６個の別個の配列が、すべての成分および鋳型、Ｔ４ポリヌクレオシドキナーゼ（成分をリン酸化する）、およびＴ４リガーゼ、ＡＴＰ、および他の適切な反応試薬を含む１つの反応によって、多重化方式で組み立てられた。反応を、３０分にわたって３７度で、次いで１時間にわたって室温でインキュベートした。シークエンシングアダプタがＰＣＲによる反応生成物に加えられ、生成物をＩｌｌｕｍｉｎａ ＭｉＳｅｑ機器によってシークエンシングした。合計１９２９１０の組み立てられた配列読取りからの別個の組み立てられた各配列の相対的なコピー数が示されている。この方法の他の実装形態は、２本鎖成分を使用することができ、成分は最初に融解されて１本鎖の変種を形成し、これをステープルにアニーリングすることができる。この方法（すなわち、ＴＤＬ）の他の実装形態または派生物を使用して、積方式で実現することができるものより複雑な識別子の組合せ空間を構築することもできる。ゴールデンゲートアセンブリ、ギブソンアセンブリ、およびリガーゼ循環反応アセンブリを含む様々な他の化学的実装形態を使用して、積方式に従って識別子を構築することもできる。

【０１３３】

核酸配列における効率的なデータ記憶のためのデータ構造およびデータブロック

【０１３４】

本章では、ＤＮＡ内にデータを効率的に符号化するためのシステムおよび方法について論じており、これは図１１～図１９に関連している。特に、特定のデータ構造および制御方式について説明する。データ構造は、記憶されたデータへの効率的なアクセスおよび修飾を可能にするデータの組織化、管理、および／または記憶のための形式を含む。より厳密には、データ構造は、１群のデータ値、これらの間の関係、およびデータに適用することができる関数または演算を含む。例示的なデータ構造および符号化方式は、全体として参照により本明細書に組み込まれている、２０１９年８月５日出願の"SYSTEMS AND METHODS FOR STORING AND READING NUCLEIC ACID-BASED DATA WITH ERROR PROTECTION"という名称の米国特許出願第１６／５３２，０７７号に記載されている。これらの構造または方式によってＤＮＡを符号化することによって、記憶されたデータにより容易にアクセスしまたはそれを操作することができる。

【０１３５】

上記で論じたように、ストリングは、ストリング内の１ビットまたはシンボルストリング内のシンボルの位置を表す識別子のライブラリとして符号化することができる。しかし、「コードワード」および「コードブック」を使用することによって、余分の変換層を用

いることもできる。「コードワード」は、順序付けられた識別子ライブラリ内のｎ$_c$個の連続する識別子のグループであり、ｎ$_c$個の利用可能な識別子からｋ$_c$個の識別子のみを設定することによって、シンボル（たとえば、任意のアルファベット）を符号化する。ｋ$_c$の値は、コードワードの「重み」と呼ばれる。「コードブック」は、すべての可能なコードワードのセットである。たとえば、本明細書に記載するシステムおよび方法は、８つの識別子のすべての隣接グループで６ビットのデータを符号化するコードブックを使用することができる。この例では、コードブックは、６ビットの可能な各ストリングを、ｎ$_c$＝８個の識別子からのｋ$_c$＝４個の固有のサブセットにマッピングすることができる（８ｃｈｏｏｓｅ４＝７０個のそのようなサブセットが存在するため、最大でｆｌｏｏｒ（ｌｏｇ$_2$（７０））＝６ビットのデータを記憶することが可能である）。これらの識別子の組合せをコードワードと呼び、コードワードが符号化するデータをワードと呼ぶ。データ内の隣接するワードは、論理的に順序付けられた識別子の中で、隣接するコードワード内に記憶することができる。コードワードは、ビットストリングとしてシンボルによって表すことができ、すべてのビット位置は、順序付けられた識別子に対応し、ビット値「０」は、コードワード内の対応する識別子の不在を表し、ビット値「１」は、コードワード内の対応する識別子の存在を表す。コードブックは、「低い重み」または「高い重み」とすることができ、たとえば低い重みのコードブックは、２５個の識別子から２つを設定することができ、高い重みのコードブックは、３５７個の識別子から１７８個を設定することができる。高い重みのコードブックは、より高密度のデータ記憶を可能にすることができる（すなわち、１つのライブラリ当たりより多くのビットが記憶される）が、低い重みのコードブックは、エラー補正方式のより良好な実装およびより頑強なエンコーダを可能にすることができる。

【０１３６】

図１１は、例示的な実装形態によるトライデータ構造から構築された識別子配列に従って反応コンパートメントに溶液を分注することによる識別子ライブラリ内のストリングの符号化の図を示す。トライは、順序付けられた木データ構造であり、トライの各層／レベルは、識別子の層を表し、各トライ層の各縁部は、対応する識別子層内の成分を表す。トライの最後の層は、識別子の多重化層を表すことができる。シンボルストリング１１００は、ＤＮＡ識別子核酸分子に記憶されるコードワードのセットを表す。コードワードは、ソースワードと呼ばれるソースアルファベットからの特有のシンボルストリングを表すシンボルストリングである。コードは、符号化として公知のプロセスにおいて、ソースワードをコードワードにマッピングする。個々の反応コンパートメント（たとえば、コンパートメント１１０６）に、液滴１１０４が分注される。シンボルストリング１１００は、木（図示せず）の根を、シンボル値「１」を指す最後の（多重化）層１１０２の葉に接続するトライパスによって表される識別子を構築することによって符号化される。

【０１３７】

概して、コードワードの重みは、コードワード内のビットの数に対する「１」の値を有するビットの数を含む。図１１に示し、特に図１５に関連してさらに後述するように、コードワードの「重み」は、コードワードにわたって均一に分散される。なぜなら、シンボルストリング１１００は、５つのビットの各ストリング（たとえば、サブストリング）が反応コンパートメント内で符号化され、ちょうど３つの「１」の値（５つから）を有するように分けられるからであり、したがって各コンパートメントは、多重化層から３つの成分を受け取って、３つの識別子を形成する。たとえば、反応コンパートメント１１０６は、多重化層に示すように、シンボルストリング「１０１０１」の位置に対応して、それぞれ塩基層からの成分（たとえば、多重化層までのトライのパスを含む成分、成分７、Ｂ、Ｄ、．．．）および多重化層からの成分０、２、および４の固有の組合せを有する識別子を収容するように構成される。「１」の値の各々に対して、そのシンボル位置に対応する識別子分子（そのシンボル位置につながるパスを構成する成分を含む）が、反応コンパートメント内へ堆積させられる。

【０１３８】

上述したように、シンボルストリングの書込みプロセスからの出力は、長期の保存および頻繁でないアクセスを必要としうる符号化されたＤＮＡ（識別子）のライブラリである。符号化されたＤＮＡの作製されたプールは、各識別子配列の多数（たとえば、数十万）の分子を収容する。グラムの場合、作製される物質の総量は、マイクログラム量とすることができる。プールは、冗長性、格納、およびアクセスのために十分な物質が存在することを確実にするために、ＰＣＲによって増幅させることができる。増幅後、プールを複数のコンテナに割り当てて、異なる場所に保存することができる。プールは、様々な核酸保存および格納システム内に保存することができる。たとえば、ＤＮＡは、冷凍庫内のＥｐｐｅｎｄｏｒｆチューブに保管することができ、液体窒素に冷凍保存することができ、またはＴｒｉｓ－ＥＤＴＡに保管することができる。ＤＮＡの貯蔵寿命は、異なる温度などの加速安定性条件にかけられた物質を読み取ることによって評価される。本明細書に記載するシステムおよび方法は、保存されたＤＮＡの長期の保存およびランダムアクセスの両方を可能にする自動化されたサンプル管理システムを含むことができる。

【０１３９】

いくつかの実装形態では、オペレーティングシステム（ＯＳ）は、エクサバイトサイズまでスケーラブルなアーカイブの書込み、読取り、発見可能な照会、またはこれらの任意の組合せを統合することが可能である。ＯＳは、記憶媒体を１群の固定サイズの「ブロック」として表すように構成することができる。各ブロックは、識別子の単一のプール内に記憶された単一の識別子ライブラリ内の識別子配列のうちの隣接配列である。しかし、ブロックは、障害許容度のためにいくつかのプール内に鏡映を作ることもできる。ＯＳは、ブロックの組織化、割当て、および書込みを担うことができる。「ブロック索引」は、「ブロックＩＤ」を物理アドレス（コンテナおよび識別子から構成される）にマッピングする階層的なデータ構造であり、「ブロックＩＤ」は、各ブロックに割り当てられた論理アドレス、バーコード、またはタグである。物理アドレスは、その対応するブロックにアクセスするために必要とされる情報を収容する。具体的には、いくつかの実装形態では、ＯＳは、上述した読取り／書込みプラットホームに関して最適化されたコーデックを介して、意味的に注釈および索引付けされたブロックの木の読取りおよび書込みを可能にする。ＯＳは、インジェストＡＰＩを含むことができる変換スタック、ならびに長期であるが粒状のデータ照会および発見のためにデータを組織化および形式化するためのモジュールを含む。ＯＳのこれらの態様は、任意の書込み、読取り、またはアクセス方法に広く適合させることができる。ＯＳの他の態様も、情報の書込み、アクセス、および読取りの方法を特別に最適化するように設計することができる。これらは、データの圧縮およびエラー保護のためのモジュール、ならびに上述した書込みシステムへのデータの構成および送信のためのモジュールを含む。上記の方法によってＤＮＡ分子に書き込まれたデータは、任意のシーケンサによって可読であるが、特有の読取り方法は以下に説明する。ＯＳはまた、たとえば情報のエクサバイトに対応することが可能な保管コンテナのシステムにＤＮＡを割り当てること、そのようなシステムからＤＮＡにアクセスすること、およびそのようなシステム内にＤＮＡを補充することによって、ライタとリーダとの間のＤＮＡに基づく情報の取扱いを仲介する自動化ソフトウェアおよびワークフローを含むことができる。

【０１４０】

いくつかの実装形態では、すべてのブロックが、固定された均一のサイズを有する。各々関連付けられたブロックＩＤを有する複数のブロックの場合、ブロックＩＤは、たとえば順序付けられた値または表現を有することによって、順序付けることができる。ブロックＩＤの順序付けは、順序付けられたブロックＩＤを有する各ブロックの物理アドレスを暗示的に示すように構成することができる。物理アドレスは、その順序付けられたブロックＩＤに対応する順序付けられた物理アドレスに各ブロックが位置付けられるように、順序付けることができる。たとえば、第１のブロックＩＤは、値または表現「０」を有することができ、第２のブロックＩＤは、値または表現「１」を有することができる。第１のブロックＩＤは、対応する第１のブロックが第１のアドレス空間内にあることを示し、第２のブロックＩＤは、対応する第２のブロックが第２のアドレス空間内にあることを示す。

【０１４１】
たとえば、識別子の組合せ空間を、固定サイズｃのコードワードの隣接するブロックに区画化することを選択することができる。このようにして、識別子がブロックの第ｇのインスタンスを符号化する範囲を、コードワードインスタンス（ｇ－１）＊ｃ＋１～ｇ＊ｃを符号化する範囲として推測することができる。このとき、共通の指定の成分セットを共用する識別子のみを取り出す化学的アクセスプログラムによって、これらの特定の識別子に容易にアクセスすることができる。これらのアクセスプログラムは、たとえばプライマーおよびＰＣＲまたは親和性タグ付きオリゴヌクレオチドおよび親和性プルダウンアッセイによって、プローブを有する前記指定の成分セットを有する識別子を選択的に標的とし、後に増幅または選択することによって機能する。プローブは、一連の選択反応に適用することができ、各反応が個々の成分を標的とする。１つの反応からの出力を、別の反応への入力として使用することができる。識別子は、成分によって論理的に順序付けることができ、さらにブロックはこれらの順序付けられた識別子の連続する範囲によって表すことができるため、ブロックを含む識別子は、異なる場合より、共通の成分を共用する可能性が高い。これにより、これらの識別子を取り出すために必要とされるアクセスプログラムの複雑さが低減される。共通の成分セットを排他的に共用する識別子の範囲にブロックが割り当てられた場合、複雑さの低減をさらに改善することができる。
【０１４２】
いくつかの実装形態では、１次シンボルストリングが、１次ストリングを複数のサブストリングに分けることによって、複数のブロックにわたって記憶される。各ブロックに記憶されたシンボルストリングは、１次シンボルストリングから取得されたシンボルの１つのサブストリングである。サブストリングは、順序付けられた形で１次ストリングを構成する。したがって、各サブストリングを記憶するブロックは、１次ストリング内の対応するサブストリングの位置または順序に従って順序付けられたブロックＩＤを有することができる。たとえば、１次ストリングを５つのサブストリングに分け、５つのブロックにわたって記憶することができ、各ブロックは、１～５の値のブロックＩＤを有する（すなわち、５つのサブストリングのうちの第１のサブストリングが、ブロックＩＤ「１」を有するブロックに記憶される）。
【０１４３】
図１２は、例示的な実装形態による機能層に組織化されたＯＳによって管理される能力の層状の組織化を示し、機能層のいくつかは、データブロックおよび／またはデータ構造を伴う。各層は、以下のリストに要約するように、層によって提供されるサービスを利用する。７つの層が、以下の設計および構造を含む６つの展開領域に変換される。
（１）コーデック：ライタ特有の最適化を伴うエンコーダ／デコーダパイプライン
（２）化学インターフェース：ビット演算から化学動作への変換
（３）自動化インターフェース：自動化デバイスへのインターフェースおよびトランスレータ
（４）ブロック抽象化：ブロックに基づくインターフェースおよびコアデータ構造への対応
（５）探索および索引：意味注釈および索引付けのためのインフラストラクチャ
（６）アーカイブアプリケーション：ＯＳを実証するアーカイブアプリケーション
本明細書に記載する符号化方式およびＯＳの利益には、書込み速度、書込みコスト、読取りコスト、またはアクセスコストに関して最適化された符号化方式を選択する能力、復号されるフットプリントを最小にするようにブロックへの索引データのマッピングを最適化する能力、大きいブロックから単一のビットおよびモデルデータ構造まですべてのスケールで情報を固有に操作する能力、ならびにデータおよび関係に関する格納、照会、および推論を可能にする現在のアーカイブ規格および慣行による緊密な統合が含まれる。
【０１４４】
コーデックは、情報のためのエンコーダ／デコーダとして機能する。上記の層はコーデックを必要とし、以下の層はコーデックがなければ有意に試験することができないため、

コーデックの適切な演算は非常に重要である。コーデックは、ソースビットストリームを受け取り、化学的方法を使用して書込みに好適な形式に変換することを担う。ソースビットストリームは、パケットに分けられ、すべてのパケットは固定のサイズである。パケットは、独立して処理することができ、並列処理のための単位として働くことができる。パケットは、１つまたは複数のブロックから構成される。ブロックは、アーカイブにおける割当てのための最も小さい単位であり、アーカイブの組合せ空間は、ブロックと呼ばれる一連の隣接ビットストリングに分けられる。固定層は、標準的な暗号ハッシングアルゴリズムを使用してブロックハッシュを算出することを担い、このハッシュは親ブロック内に含まれる。ブロックが復号されるとき、その完全性は、そのハッシュを再算出し、親ブロックを介してそのハッシュを検査することによって検査することができる。

【０１４５】

いくつかの実装形態では、コーデックは、図１３に示す例示的な符号化方式を含めて、Ｂ木構造またはトライ構造などのデータ構造に従って、符号化を実行または調整する。

【０１４６】

図１３は、例示的な実装形態による識別子の設計および順序付けのための層状デカルト積の組合せコンストラクタ（ＬＣＰＣＣ）のシステム図である。プロダクトコンストラクタは、３つの層（Ｍ＝３）および８つの成分配列の成分ライブラリ（Ｃ＝８）を有する。層ごとに｛３，３，２｝の成分配列を有する組合せの区画方式が示されており、これは第１の層内に３つの成分、第２の層内に３つの成分、および第３の層内に２つの成分があることを意味する。すべての可能な識別子配列の空間が、組合せ空間を形成する。成分ライブラリから構築可能な組合せオブジェクトの総数は、組合せ方式のスパンと呼ぶことができ、識別子の単一のプールに書込み可能なビットストリームの長さを決定する。この特定の方式のスパンは、３×３×２または１８である。概して、任意の組合せ区画方式を使用することができ、Ｃ個の成分がＭ個の層に分離され、ここで第ｉの層はＣ$_i$個の成分を有し、Ｃ$_i$のＭ個の値の和がＣであり、Ｃ$_i$のＭ個の値の積は、可能な識別子配列の数、したがって書込み可能なビットストリームの長さを画定する組合せ空間のスパンである。組合せオブジェクトは、成分配列のランク付けを、そこから構築される識別子まで延ばすことによって、組合せ空間内で辞書的に順序付けられる。この順序付け情報は、識別子内で暗示的であり、組合せ空間内のその位置を識別し、ソースシンボルストリーム内で識別子によって符号化されたシンボルの位置を識別するために使用することができる。たとえば、ＬＣＰＣＣを使用して、２進アルファベットを符号化し、構築された識別子によって符号化されたシンボルを「１」になるように画定することができる。「０」のシンボルは、対応する識別子を構築しないことによって表すことができる。ソースビットストリームは、そのビットストリームに固有の特有の識別子セット（すなわち、識別子ライブラリ）を構築することによって符号化することができる。

【０１４７】

図１４は、例示的な実装形態によるデータブロックのマッピングおよびデータ構造の使用のためのアーカイブ演算のシステム図を示す。アーカイブ（ＣＡＲ）は、ブート、オントロジ、索引、およびコンテンツの領域に区画される。ブート区画は、外部メタデータなく復号することが可能な標準的な符号化方式を使用して書き込むことができ、他の区画を読み取るために必要とされるパラメータ、鍵、およびアドレスを記憶することができる。ＯＳは、上述したように、記憶媒体を１群の固定サイズのブロックとして抽象化する。

【０１４８】

各ブロックは、単一の識別子プールとして記憶された単一の識別子ライブラリ内に識別子配列の隣接配列を含むことができ、障害許容度のためにいくつかの識別子プール内に鏡映を作ることができる。概して、ＯＳは、ブロックの組織化、割当て、および書込みを担う。ブロック層がソースビットストリームパケットを受け取るとき、ブロック索引は、パケットを分けてアーカイブ内のブロックに割り当てる。ブート区画は、ブロック索引を含み、階層データ構造が、ブロックＩＤを物理アドレス（コンテナおよび識別子から構成される）にマッピングする。ブロック索引は、自由／使用中のブロックを追跡し、新しいパ

ケットにブロックを割り当てる。

【０１４９】

各ブロックＩＤは、論理アドレスとすることができ、分子アーカイブ内の物理アドレスに変換することができる。これは、図１４に示すようにブロック索引を横断することによって実現される。データ構造内の各ノード（たとえば、アドレスノード）は、Ｂ木データ構造に類似した子ブロック識別子範囲の配列を含むことができる（たとえば、図１５に関連して後述）。各範囲は、関心ブロックへのパス上の次のブロックを指す。このようにして、システムは、アドレスノードの木を維持し、この木は、実際のデータを収容する分子アーカイブ内のブロックを参照する葉ノードで終わる。言い換えれば、葉ノードは、ブロックの物理アドレスを識別するブロックＩＤを記憶しており、ブロックのハッシュをさらに記憶することができる。内部ノードはまた、その子ノードのハッシュの連結などのハッシュを収容し、したがってハッシュ木を形成することができる。図１４に示すように、ブロックの物理アドレスは、場所コード、コンテナコード、ならびに識別子範囲を含むことができ、識別子範囲は、関連する情報を符号化する複数の識別子を参照する開始および終了識別子によって画定される。概して、障害許容度を可能にするために、ブロックＩＤを２つ以上の物理アドレスに分解することが可能である。この場合、ブロックＩＤに記憶された情報は、２つもしくはそれよりも多い異なるコンテナまたは２つもしくはそれよりも多い異なる識別子範囲に分散させることができる。

【０１５０】

ビットブロック上の各高レベル演算は、化学的方法または物理的ステップに依拠する複数の物理的動作に依存し、そのような複数の物理的動作をもたらす。それらの化学的または物理的プロセスを調整するために、２つのタイプのソフトウェアツールを使用することができる。第１に、最適化ツールが、ブロック演算を、物理的動作の最適化されたセットに変換する。次いで、変換ツールが、物理的動作を、技術者または自動化デバイスによって実行されるべき詳細な行動プログラムに変換し、これは、ビットブロックの演算と物理的および化学的動作との間でトランスレータを設計および実施することを含むことができる。

【０１５１】

本明細書に記載するシステムおよび方法は、アーカイブの保存ならびに標的の発見および照会を提供する。本開示は、特有の標的アクセス演算を実行するようにデータブロックおよびデータ構造に影響を与えることが重要であり、これはアーカイブの大部分の復号を必要としない。代わりに、本開示のシステムおよび方法によって、標的コンテンツの選択的かつ増分的な発見、照会、および読取りを行いながら、アーカイブ上の結合演算および他の構造を算出する必要を最小にすることが可能である。最小にされるべき主なメトリクは、照会の配列を満たすために復号されるビットの総数である。

【０１５２】

図１５は、例示的な実装形態によるコンテナ内のブロック識別（ＩＤ）に関連付けられたデータのブロックを記憶するステップについて概説する流れ図１５００を示す。ステップ１５０２で、複数のブロックが取得される。各ブロックは、シンボルストリングを含み、ブロックＩＤに関連付けられる。ブロックＩＤは、特定のブロックに関連付けられた任意の識別特性またはシンボルとすることができる。たとえばブロックＩＤは、トリプルの形式の意味注釈とすることができる。いくつかの実装形態では、ブロックＩＤは、整数、ストリング、位置、トリプル、属性のリスト、または意味注釈である。たとえば、ブロック内に含まれるシンボルストリングの第１のＸのシンボルは、そのブロックに対する数値ＩＤを示すことができる。

【０１５３】

ステップ１５０４で、ブロック（ステップ１５０２で受け取った複数のブロックに属するブロックのうちの１つ）がコンテナに割り当てられる。コンテナは、核酸分子を記憶することができるビン、チューブ、または他の物理的記憶媒体などの物理的な場所とすることができる。コンテナは、単一のブロックまたは複数のブロックに結合することができる

。たとえば、１つのコンテナをＢ個の情報ブロックに関連付けることができる。いくつかの実装形態では、コンテナは、複数のサブコンテナを含むことができる。

【０１５４】

ステップ１５０６で、ブロックは、コンテナに関連付けられるべき識別子配列にマッピングされる。これらの識別子は、識別子範囲または識別子範囲の複数の異なる識別子を含む。識別子範囲は、範囲を取り囲む識別子を含む成分配列によって指定される。いくつかの実装形態では、個々の各識別子は別個の整数に関連付けられており、したがって識別子範囲を２つの整数によって指定することができる。複数の識別子配列のうちの個々の識別子配列は、ブロックに記憶されたシンボルストリング内の個々のシンボルに対応する。各識別子配列は、対応する複数の成分配列を含む。これらの成分配列の各々は、別個の核酸配列を含む。

【０１５５】

ステップ１５０８で、複数の識別子配列の個々の識別子が構築される。たとえば、Ｑ個の識別子配列のセットが、特定のコンテに関連付けられる。それらＱ個の識別子配列のサブセットＶは、上述した様々な方法に説明したように、ブロック内の情報を表すように物理的に構築することができる。ステップ１５１０で、ステップ１５０８で構築された識別子は、割り当てられたコンテナ内に記憶される。たとえば、割り当てられたコンテナはこのとき、ブロック内に記憶された情報を表す数Ｖの識別子を保持する。コンテナおよびコンテナに関連付けられた複数の識別子核酸配列の識別情報は、関連付けられたブロックＩＤを使用して判定されるように構成される。いくつかの実装形態では、識別情報は、関連付けられたブロックＩＤを使用した各コンテナの識別情報へのアクセスを容易にするように設計されたデータ構造に記憶される。たとえば、データ構造は、Ｂ木、トライ、またはアレイのうちの１つである。いくつかの実装形態では、データ構造の少なくとも一部分は、デジタル情報とともに索引内に記憶される。索引は、第２のコンテナに関連付けられた第２の複数の識別子配列を含む。いくつかの実装形態では、索引は、磁気記憶デバイス、光記憶デバイス、フラッシュメモリデバイス、またはクラウドストレージ内に記憶される。

【０１５６】

いくつかの実装形態では、索引は、Ｂ木データ構造を含む。この場合、Ｂ木の各ノードは、第２の複数の識別子配列の別個の複数の識別子（すなわち、ステップ１５０８で構築された識別子のセットとは異なる）を含むことができる。プロセスは、Ｂ木を探索して別個の複数の識別子の識別情報を判定することを伴う。具体的には、Ｂ木内の特定のブロックＩＤを探索することは、第１のノードを含む別個の複数の識別子を選択することと、第１のノードの値を読み取ることとを伴う。識別子を選択してノードの値を読み取るステップは、後続のノードに対して繰り返すことができる。後続のノードを含む別個の複数の識別子の識別情報は、第１のノードの値に関連するブロックＩＤによって判定される。一例では、第１のノードはＢ木の根ノードであり、ノードを選択して読み取るプロセスは、Ｂ木の葉ノードの値が読み取られるまで継続する。葉ノードの値は、ブロックＩＤに対するブロックが存在するかどうかを通信するように構成される。ブロックＩＤが存在する場合、前記ブロック（たとえば、識別子範囲）を含むコンテナの識別情報および複数の識別子核酸配列の識別情報を、ユーザまたはシステムへ通信することができる。

【０１５７】

いくつかの実装形態では、索引はトライデータ構造である。この場合、トライの各ノードは、第２の複数の識別子配列の別個の複数の識別子を含むことができる。いくつかの実装形態では、ブロックＩＤはシンボルストリングであり、トライ内の各ノードは、シンボルストリングの可能な接頭辞に対応する。特定のブロックＩＤに対するトライを通るパスが存在する場合、対応するブロックの物理アドレス（コンテナおよび１つまたは複数の識別子範囲から構成される）を、そのパスの葉ノードによって指定することができる。トライの各中間ノードは、別個の複数の識別子によって表すことができ、どれだけの娘ノードを有するか、それらの娘ノードがどのシンボルを表すか、ならびにそれらの娘ノードの物理アドレス（コンテナ識別情報および１つまたは複数の識別子範囲から構成される）に関

する情報を収容することができる。そのようにして、Ｂ木と同様に、本明細書に記載するアクセスおよび読取り演算を使用して、ＤＮＡ内でトライをナビゲートすることができる。方法１５００は、第２の複数の識別子配列を含む識別子のプールから物理アドレスにアクセスし、そのようなプールからの一連のプローブを使用することをさらに含むことができる。

【０１５８】

いくつかの実装形態では、データ構造はアレイである。この場合、アレイの各要素は、第２の複数の識別子配列の別個の複数の識別子を含む。アレイ内の各要素は、ブロックＩＤに対応することができ、そのブロックＩＤの物理アドレス（コンテナ識別情報および識別子範囲を含む）を収容することができる。

【０１５９】

物理アドレスは、物理アドレスを追加のデータ構造に記憶することなく、ブロックＩＤが物理アドレスにマッピングされるように、ブロックＩＤに固有に構成することができる。たとえば、ブロックＩＤは、物理アドレスに関連付けられた複数の識別子配列のうちのすべての識別子配列によって共用される複数の成分配列にマッピングされる。ブロックに関連付けられた複数の識別子配列は、隣接して順序付けられた識別子核酸配列（たとえば、図１３に関連して説明）を含むことができ、したがって前記複数の識別子配列は、範囲の第１および最後の識別子の識別情報を含む識別子範囲によって、対応する物理アドレス内に指定される。第１および最後の識別子は、整数によって表すことができる。一連のプローブを使用して、複数の識別子（たとえば、ブロックに関連付けられたもの）にアクセスすることができ、一連のプローブは、連続することができる。プローブは、ＰＣＲプライマーとすることができ、したがってアクセスはＰＣＲを介して実行され、またはプローブは、親和性タグ付きオリゴヌクレオチドとすることができ、したがってアクセスは親和性プルダウンアッセイを介して実行される。

【０１６０】

ブロックＩＤが位置である実装形態では、前記位置は、親シンボルストリングの対応するブロックによって表されるシンボルストリング内の位置とすることができる。前記親ストリングは、たとえば別のシンボルストリングにおけるパターンの発生の計数または位置付けのためのデータ構造を含む。以下で論じるように、前記データ構造は、Ｂｕｒｒｏｗｓ－Ｗｈｅｅｌｅｒ変換（ＢＷＴ）、接尾辞アレイ、接尾辞木、または転置索引とすることができる。

【０１６１】

データ構造は、接尾辞木に基づく手法を使用して、データにおけるパターン発生の判定、位置付け、および計数を助けるために、ブロック内に記憶することができる。接尾辞木において、ブロックは、接尾辞木のノードを表すように構成される。接尾辞木はトライであり、根ノードから葉までのすべてのパスが、シンボルストリングＳの接尾辞を表す。トライの縁部は、各パスを含むシンボルのサブストリングを表す。接尾辞木は、識別子核酸ライブラリ内に表すことができ、すべてのブロックが、接尾辞木内のノードに対応し、その娘ノードについての情報を収容する。たとえば、娘ノードについての情報は、各娘ノードにつながる縁部を含むシンボルのサブストリング、およびそれらの娘ノードを収容するブロックの物理アドレスを含む。根ノードは、第１のブロックのような所定のブロックとすることができる。パターンのメンバーシップ、カウント、または場所を照会することは、根ノードに対応するブロックの識別子にアクセスし、その中に収容されていた情報を復号し、その中に収容されていた情報および照会パターンに基づいて、次のブロックの物理アドレスを判定し、対応する照会を満たす下流のブロック（またはノード）がなくなるまで、または先導ノードに到達するまで、プロセスを継続することによって、接尾辞木に沿ってパスをたどることを伴う。前者の場合、ストリングＳ内に照会パターンは存在しない。後者の場合、葉ノードに対応するブロックが、照会パターンのカウントまたは場所を収容するように構成することができる。

【０１６２】

データ構造は、転置索引に基づく手法を使用して、データにおけるパターン発生の判定、位置付け、および計数を助けるために、ブロック内に記憶することができる。転置索引では、シンボルストリングＳを含むシンボルアルファベットに、固定長の可能な各サブストリングに対するブロックが存在することができる。各ブロックは、Ｓ内の対応するサブストリングの開始位置についての情報を収容することができる。ブロックＩＤは、サブストリングに対応することができ、または分類されたサブストリングの位置に対応することができ、したがって追加の情報なしで、サブストリングに対応するブロックの物理アドレスを確かめることができる。照会パターンは、それを含む転置索引のサブストリングにマッピングすることができ、ブロックにアクセスして復号することができる。その中に収容されている位置情報を使用して、Ｓ内の照会パターンのカウントおよび場所を判定することができる。転置索引は、固定長のサブストリングに限定される必要はない。たとえば、転置索引は、１つの文書内または複数の文書にまたがるワードの位置を表すために使用することもできる。

【０１６３】

データ構造は、微小空間におけるフルテキスト索引（ＦＭ索引）に基づく手法を使用して、データにおけるパターン発生の判定、位置付け、および計数を助けるために、ブロック内に記憶することができる。ＦＭ索引は、Ｂｕｒｒｏｗｓ－Ｗｈｅｅｌｅｒ変換（ＢＷＴ）に基づくサブストリング索引であり、接尾辞アレイに類似している。ＦＭ索引は、入力データまたはテキストの圧縮を可能にしながら高速のサブストリング照会も可能にするデータ構造である。ＦＭ索引の構造について、以下でより詳細に説明する。ＦＭ索引は、圧縮されたデータ／テキスト内のパターンの発生の数を効率的に発見し、ならびに各発生の位置を位置付けるために使用することができる。したがって、いくつかの実装形態では、ブロック内に記憶されたシンボルストリングは、第２のシンボルストリング内の任意のシンボルサブストリングのメンバーシップの位置付け、計数、および／または判定に対応することが意図されたデータ構造の一部分であり、第２のシンボルストリングは、第１のストリングより大きくすることができる。データ構造は、ＦＭ索引、カウンタアレイ、Ｂｕｒｒｏｗｓ－Ｗｈｅｅｌｅｒ変換、または接尾辞アレイとすることができ、これらのデータ構造の各々について、図１６～図１９を参照して以下でさらに詳細に論じる。いくつかの実装形態では、データ構造は、たとえば第２のストリングが記憶されたコンテナとは別個の、１つまたは複数のコンテナの別個のセット内に記憶される。

【０１６４】

上述したように、データ構造は、ＢＷＴを含むことができ、ＢＷＴは、ストリングＳをストリングｂｗ（Ｓ）の変換に変換する。変換は、ストリングＳの探索を支援する特定の特性を有する（図２６～図３２に関連する下記の説明参照）。概して、ＢＷＴは、変換されていないストリングＳに対してより容易に実施することができる方法を介した圧縮に好適な方法によって、変換内で順序付けられた入力シンボル／ビットの置換または無損失変換であると考えることができる。ＢＷＴは、入力ストリングＳのシンボルを再配置してｂｗ（Ｓ）を形成する順方向変換、およびそのＢＷＴｂｗ（Ｓ）から元のストリングＳを再構築する逆方向変換という１対の変換を含む。

【０１６５】

概して、長さｎを有する入力ストリングＳ＝$s_1$、$s_2$、．．．、$s_n$の場合、Ｓのシンボルが、順序付けられたアルファベットΣ（たとえば、英語のアルファベット、正の整数のセット、任意の文字もしくはシンボルのセット、またはこれらの組合せ）から選択される。順方向変換は、この入力ストリングＳに関して次のように進行する。ストリングｓ＄が構築され、ここで＄は、順序付けられたアルファベットΣ内に生じない特別なシンボルであり、その全体的な順序付けに従ってアルファベット内のあらゆる他のシンボルより小さいと仮定される。たとえば、英語のアルファベットの場合、全体的な順序は、＄、Ａ、Ｂ、Ｃ、．．．Ｘ、Ｙ、Ｚになるはずである。概念上、サイズ（ｎ＋１）×（ｎ＋１）の行列Ｍは、ストリングｓ＄の循環左シフトのすべてである行を収容する。行列Ｍは、ｓの回転行列と呼ぶことができる。ストリングの左シフトは、ストリングの第１のシンボルを

ストリングの末端へ動かすことと、すべての他のシンボルを１つの位置だけ左へシフトさせることとを伴う。この左シフトは、ｓ＄のすべての可能な循環左シフトがＭに含まれるまで繰り返される（図１６の左側参照）。行列Ｍは、Ｂｕｒｒｏｗｓ－Ｗｈｅｅｌｅｒ変換中に明示的に構築される必要はない。次いで、アルファベットΣに定義される順序に従って、＄は順序の最初であると考えて、行を左から右へ読み取って、Ｍの行を辞書的に分類する。＄はΣ内のあらゆるシンボルより小さく、ストリング内に一度だけ現れるため、最後に分類される行列Ｍ’は、＄ｓの第１の行を含む（図１６の右側参照）。行列Ｍ’の最後の列は、＄を含む列Ｌに対応し、Ｂｕｒｒｏｗｓ－Ｗｈｅｅｌｅｒ変換ｂｗ（ｓ）は、（Ｌ’，ｒ）に等しく、ここでＬ’は、Ｍ’の最後の列を読み取ってシンボル＄を省略することによって取得されるストリングであり、ｒは、最後の列におけるシンボル＄の位置である。

【０１６６】

図１６は、例示的な実装形態によるストリングＳ＝「ａｂｒａｃａｄａｂｒａ」に対するＢＷＴの例示を描く。分類されていない１２×１２の行列Ｍが左に示されており、Ｓ＄の各循環左シフト有し、Ｓ内の１１個のシンボルおよび追加のシンボル＄に対して１２行をもたらす。図１６の右側は、分類された形の１２×１２の行列Ｍ’を描いており、＄はＳ＄のすべてのシンボルの順序の最初であると定義され、ストリングの残りは英語のアルファベットの順序に従って分類されるため、＄から開始するストリング「＄ａｂｒａｃａｄａｂｒａ」に対応する第１の行を有する。シンボルａから開始する５つの行（「ａｂｒａｃａｄａｂｒａ」というワード内に文字ａが５回現れるため）が、後続のシンボル（たとえば、第２、第３、第４、および第５のシンボル）に従ってアルファベット順に分類される。行列Ｍを変換してＭ’を導出するこのプロセスを、接尾辞分類または右分類と呼ぶことができる。

【０１６７】

Ｆによって示されている分類された行列Ｍ’の第１の列を読み取ることで、Ｓ内のすべてのシンボルの分類された配列であるストリング「＄ａａａａａｂｂｃｄｒ」が与えられる。第１の文字＄を含まないが同じストリングが、Ｆ’または「ａａａａａｂｂｃｄｒ」として表される。出力ストリングＬ’は、最後の列Ｌ（「ａｒｄ＄ｒｃａａａａｂｂ」）を読み取り、シンボル＄を除外することによって取得され、したがってＬ’は「ａｒｄｒｃａａａａｂｂ」に等しい。最後の列におけるシンボル＄の単一の発生の位置は、ｒ＝３と示されている。出力ストリングＬ’は、局所的に均質な特性を有し、これは圧縮設定に特に有用である。具体的には、局所的に均質な特性は、図１６の最後の列Ｌの最後の６つのシンボルが、圧縮方法を介して大幅に圧縮することができる非常に反復性の高いストリング「ａａａａｂｂ」を形成することから明らかである。局所的均質性のこの特性は、ストリングが右のコンテキスト（すなわち、接尾辞）に従ってアルファベット順に分類されるため、ＢＷＴに固有である。

【０１６８】

本明細書では、接尾辞アレイ（ＳＡ）を使用することができる。長さｎのシンボルのストリングに対して構築された接尾辞アレイｓａ［０，ｎ－１］は、辞書的順序でｓの第ｉの接尾辞の位置をｓａ［ｉ］に記憶する。接尾辞位置ｓａ［ｉ］は、$\log_2(n)$に等しい複数のビットで符号化することができ、接尾辞アレイｓａは、２進で直列化することができる。この直列化は、接尾辞アレイ内の接尾辞位置の各ビット表現を連結すること、または各ビット表現を別個のコンテナ内の識別子に記憶することを伴う。

【０１６９】

図１７は、同じストリングのＢＷＴと比較して、例示的な実装形態によるストリングＳ＝「ａｂｒａｃａｄａｂｒａ」に対する例示的な接尾辞アレイを描く。図１７の第１の列は、シンボル＄が添えられた入力ストリングＳであるストリングＳ＄のすべての接尾辞を示し、ここでシンボル＄は、シンボルの全体的な順序の最初であると定義される。各接尾辞は、ストリングＳ＄内の各位置に対して、その位置のシンボルおよびすべての後続のシンボルを得ることによって取得される。対応する接尾辞位置、Ｓ＄内の各接尾辞の開始位

置が、図１７の第２の列にある。たとえば、上の行の第１の接尾辞は、接尾辞位置０を有し、したがってストリングＳ＄全体または「ａｂｒａｃａｄａｂｒａ＄」を有する。上の行の第２の接尾辞は、接尾辞位置１を有し、したがってストリングＳ＄のうち第１の位置の後に始まる部分または「ｂｒａｃａｄａｂｒａ＄」を有する。残りの接尾辞も同様に判定される。

【０１７０】

ＢＷＴが行を辞書的に分類することを伴うのとちょうど同様に、図１７の第１の列からの接尾辞は、アルファベット順に従って辞書的に分類され、図１７の第３の列にリスト化され、対応する接尾辞位置は、図１７の第４の列にリスト化される。ここでは、シンボル＄が辞書的順序で最初にあると定義されるため、接尾辞「＄」は常に、分類された接尾辞アレイの第１の行にある。列４における分類された接尾辞位置のリストを、物理的に記憶される接尾辞アレイとすることができる。

【０１７１】

図１７の最後の２つの列は、図１６に描くように、同じ入力ストリングがＢＷＴに変換されるため、分類された回転行列Ｍ’および対応する最後の列Ｌを示す。第３の列の分類された接尾辞は、行列Ｍ’の行に対応し、行列Ｍ’の各行に対して、対応する分類された接尾辞は、シンボル＄を含むシンボル＄までのシンボルのセットである。第ｉの行に対する最後の列Ｌ内のすべてのシンボルＬ［ｉ］が、図１７の同じ行（すなわち、第ｉの行）の接尾辞位置ｓａ［ｉ］に先行する入力ストリングｓのシンボルに対応する。しかし、接尾辞が全ストリング（すなわち、ｓａ［ｉ］＝０）である場合、＄が先行するシンボルとして使用される。このようにして、ＳＡに対して接尾辞を分類することは、ＢＷＴに対してＭの行を分類することに均等となりうる。各構造間の関係は、以下の等式によって形式化される。

【化１】

$$L[i] = \begin{cases} s[sa[i]-1] & if\ sa[i] \neq 0 \\ \$ & \text{それ以外} \end{cases} \qquad \text{等式１}$$

この関係は、回転行列Ｍ’の行とストリングｓの接尾辞との間に全単射の対応関係が存在することを示す。ストリングｓの接尾辞アレイを考慮すると、ＢＷＴに対するストリングＬを導出するには線形時間を要する。

【０１７２】

図１８および図１９は、例示的な実装形態による本明細書に記載するデータ構造を実施する実験例を描く。この実験例は、３７８成分の成分ライブラリサイズを使用して実施されたものであり、これらの成分は、以下の区画方式（３，３，３，３，３，３，３，３５７）によって示す８つのレベルの中で細分される。このライブラリに対応するトライに対するファンアウトは、最初の７つのレベルに対して値ｃ＝３、次いで最後のレベルに対して値ｃ＝３５７を要する。この成分ライブラリによって、合計７８０，７５９の固有の識別子を符号化することができる（すなわち、$3^7 \times 357 = 780,759$）。いくつかの実装形態では、たとえば手法を実行可能かつ頑強にするためのそのような識別子の書込みおよびエラー補正コードの設計に関する技術上の制約のため、索引付けされたストリングｓのシンボルを符号化するためにすべてのそのような識別子が使用されるとは限らない。図１８および図１９に描く例は、例示のみを目的として示されており、本開示は、任意の数のレベルに対する任意のサイズの成分ライブラリに適用可能であることが理解されよう。本開示では、「コンテナ」という用語は物理的コンパートメントを指すことが多いが、図１８および図１９に関連して説明するコンテナは、必ずしも別個のコンパートメント内に物理的に区画化されていない核酸分子のセットを指すことを理解されたい。

【０１７３】

図１８は、例示的な実装形態によるストリングのＢｕｒｒｏｗｓ－Ｗｈｅｅｌｅｒ変換の符号化の例示的な表現を示す。この例では、コンテナは、同じ長さ７成分の接頭辞を共

用し、最後の層に区画化された３５７個の可能な成分のうちの最後の成分に従って区別される識別子のグループを指す。同じ７「塩基」レベルを共用する識別子は、たとえば単一の自己組織化反応で識別子を多重化することによって、単一のコンテナ内に並列に構築される。$3^7=2,187$個のコンテナが存在し、各コンテナは、識別子構造のための最後の層で利用可能な３５７個の成分に従って、３５７個の固有の識別子を収容することができる。７塩基レベルが並列に構築されるとき、これらは、最後の層に別個の成分を有する識別子の数に等しい量に複製することができる。

【０１７４】

３５７個の利用可能な識別子から、１つのコンテナ当たり３５０個の識別子のみを使用する低い重みのコードブックを実施することができる。概して、低い重みのコードブックは、１つのコンテナ当たりの識別子の数が閾値より小さい限り、１つのコンテナ当たり利用可能な識別子の数に対して、１つのコンテナ当たり任意の好適な数の識別子を使用することができる。コードブックは、２５個の識別子のブロック内に３５０個の識別子を区画化し、１つのコンテナ当たり１４個のブロックが得られる。各ブロックは、コードワードに対応することができる。各コンテナは、数ビットの情報を符号化する１２個のデータコードワード、ならびにエラー補正コードを実施する２個のエラー検出および補正（ＥＤＡＣ）コードワードに区画化することができる。各コードワードが２５個の利用可能な識別子から２つの識別子を設定した場合、３００個の可能なコードワード構成（２５ｃｈｏｏｓｅ２）は、１バイト（８ビット）、したがってエラー補正を伴って１つのコンテナ当たり９６個のソースビットの符号化を可能にする。$2,187$個のコンテナのセットの場合、$26,244$バイトの情報を符号化することができ、$4,374$バイトをエラー補正に使用する。

【０１７５】

$2^{13}=8,192$ビット（１キロバイト）に等しいサイズｎの２進ストリングｓを考慮すると、Ｂｕｒｒｏｗｓ－Ｗｈｅｅｌｅｒ変換をストリングｓの符号化に使用することができる。ストリングｓのＢＷＴの長さ（すなわち、ＢＷＴ（ｓ））は、ストリングｓの長さと同じであり、この例では、具体的に$n=2^{13}$である。教示法アルゴリズムを使用して、ｓのＢＷＴを構築することができる。低い重みのコードブックによれば、各コンテナは、１２バイトまたは９６ビットの情報（各データコードワードに対して１バイトまたは８ビット）を符号化する。長さ$8,192$ビットを有するｓのＢＷＴ、ＢＷＴ（ｓ）は、８６個のコンテナ（$8,192$ビットの上限を１つのコンテナ当たり９６ビットで割った値）で直列化される。サイズ$w=96$ビットのブロックは、各コンテナがＢＷＴ（ｓ）のブロックを符号化するように画定することができる。さらに、ＢＷＴ（ｓ）の１ブロック（９６ビット）当たりの特定のシンボル値のランニングカウントを記憶するカウンタアレイを符号化することができる。

【０１７６】

図１９は、例示的な実装形態による図１８のＢｕｒｒｏｗｓ－Ｗｈｅｅｌｅｒ変換から導出されるカウンタアレイを符号化する例示的な表現を示す。カウンタアレイは、長さがＢＷＴ（ｓ）内のシンボルの数に等しいサイズの数$2^{13}=8192$を記憶するのに十分な設定長さ$b=13$ビットのブロックまたはカウンタに分けられる。したがって、コンテナは、１２個のコードワードおよび追加の２個のｅｄａｃコードワードによって$floor(96/13)=7$カウンタを記憶することができる。この例では、カウンタアレイの各ブロックは、ｓのＢＷＴ、$BWT_s$内のブロックに対応し、その位置の接頭辞内に１の数を記憶する。この関係は、カウンタアレイを$c[0, floor((n-1)/w)]$として画定するように形式化され（カウンタアレイｃは長さ$floor((n-1)/w)+1$）を有する）、したがってｃ［ｉ］（所与のブロックまたはカウンタ、ｃの第ｉのブロック）が、接頭辞$BWT_s[0, i\times w-1]$内に１の数を記憶する。したがって、ｃ［０］（ｃの第１のブロック）は０に等しい。所与のカウンタｃ［ｉ］は、位置$floor(i/7)$のコンテナ内に記憶され、第ｉのコンテナの前の$BWT_s$の部分内で生じる１（ビット値１）の数を計数する。

【０１７７】
ＢＷＴ$_s$は、長さｎ＝２$^{１３}$ビットを有し、各カウンタは、１３ビットに等しいｌｏｇ$_2$（ｎ）個のビットで符号化することができる。各々１３ビットの７つのカウンタは、合計９１ビットに等しく、これは９６ビットという１つのコンテナ当たりの最大記憶より小さいため、カウンタアレイｃは、１つのコンテナ当たり７つのカウンタを割り当てることによって２進で直列化される。したがって、１３ビットで表される所与のカウンタｃ［ｉ］が、ｆｌｏｏｒ（ｉ／７）で示されるコンテナ内に記憶され、コンテナを０から計数する。８６個のコンテナが、ｓの変換、ＢＷＴ$_s$を記憶し、したがって合計８６個のカウンタがアレイｃを形成する。合計８６個のカウンタは、１３個のコンテナ（８６個のカウンタの上限を１つのコンテナ当たり７つのカウンタで割った値）にわたって記憶される。図１９は、この例示的な分布を図示する。
【０１７８】
カウンタアレイに関して上述したように、同じストリングｓに対する接尾辞アレイの接尾辞位置は、１３ビット（ｌｏｇ$_2$（ｎ）個のビット）で符号化することができる。接尾辞アレイは、１つのコンテナ当たり７つの接尾辞位置を記憶することによって、２進で直列化される。上記の同じ例の場合、所与の接尾辞位置ｓａ［ｉ］が、コンテナｆｌｏｏｒ（ｉ／７）に位置付けられ、コンテナを０から計数する。接尾辞アレイは、ｃｅｉｌｉｎｇ（ｎ／７）個のコンテナを要し、これはｃｅｉｌｉｎｇ（８，１９２／７）個のコンテナ、または１，１７１個のコンテナに等しい。
【０１７９】
上述したデータ構造は、合計１，２７０個のコンテナ（ＢＷＴに対する８６＋カウンタアレイに対する１３＋接尾辞アレイに対する１，１７１）に記憶される。こうしたコンテナの使用は、この例に使用される成分ライブラリに従って、２，１８７個の利用可能なコンテナの一部分である。全体的な標的ライブラリは、ＢＷＴライブラリＬ$_{BWT}$、カウンタライブラリＬ$_C$、および接尾辞ライブラリＬ$_{SA}$という表記に従って細分することができる。いくつかの実装形態では、この細分は、単に論理的または暗示的であり、したがってこれらのライブラリへのアクセスは並列に行うことができる。並列アクセスの場合、各ライブラリへの照会を組み合わせることができる。
【０１８０】
上記の例では、ｒａｎｋ$_1$（ｘ）、すなわちＢＷＴ$_s$のビット位置ｘを含むビット位置ｘまでのビット値「１」の発生の総数を取得することは、以下を伴う。第１に、ＢＷＴ$_s$におけるそのビット位置、すなわちブロック位置ｚ＝ｆｌｏｏｒ（ｘ／ｗ）のブロックを計算し、ここでｗ＝９６はＢＷＴ$_s$のブロックサイズである。第２に、Ｌ$_{BWT}$のコンテナ位置ｚにあるそのコンテナにアクセスすることによって、前記ブロックを読み取り、その中に符号化されたデータを復号して、位置ｘを含む位置ｘまでのビット値「１」の数の第１のカウントｎ$_{BWT}$を計算する。第３に、１つのコンテナ当たり７つのカウンタがあるため、Ｌ$_C$のコンテナ位置ｆｌｏｏｒ（ｚ／７）にあるカウンタアレイ内の対応するカウンタｚを読み取り、その中のデータを復号して、ビット位置ｘを収容するブロックまでのビット値「１」の数の第２のカウントｎ$_C$を計算する。最後に、第１のカウントおよび第２のカウントの和（ｒａｎｋ$_1$（ｘ）＝ｎ$_{BWT}$＋ｎ$_C$）を得る。ｒａｎｋ$_0$（ｘ）＝ｘ－ｒａｎｋ$_1$（ｘ）＋１であるため、この方法は、０に対するランクの計算に拡大可能である。
【０１８１】
この例では、カウンタアレイの第ｉのブロックは、ＢＷＴの第ｉのブロックまで、特定のシンボル値を計数するが、他の実施形態では、カウンタアレイの第ｉのブロックは、ＢＷＴの第ｉのブロックを含む第ｉのブロックまでの特定のシンボル値を計数することができることを理解されたい。差が上記の第３のステップのみ変化させるはずである場合、ｎ$_C$は、位置ｚではなく位置ｚ－１のカウンタに対応するはずである。別法として、これらのステップは、ブロックｚにわたってビット値「１」の総数を計数し、次いでブロックｚ内の位置ｘの後に生じるビット値「１」の数を引くように構成することができる。概して、カウンタブロックｆ（ｚ）へのＢＷＴブロックのマッピングは、方法の効率に影響を及

ぼすことなく定義することができる。この例で使用されるコンテナに属する識別子は、各々成分に結合する一連のプローブによって効率的にアクセス可能になるように構成される（各コンテナの識別子が、共通の７つの成分の排他的なセットを共用するため）ことも理解されたい。さらに、コンテナ内に収容される情報は、１つのコンテナ当たり２つの関連付けられたｅｄａｃコードワードのため、特定の許容度に補正可能になるように構成されることも理解されたい。概して、カウンタアレイおよびＢＷＴ$_{s}$の識別子への異なるマッピングを定義することができる。

【０１８２】

核酸配列に記憶されたデータの効率的な読取りおよびアクセス演算

【０１８３】

識別子にアクセスして読み取り、符号化された情報を明らかにすることができ、読取りおよびアクセス演算について、図２０～図２２に関連して説明する。識別子のライブラリＬを考慮すると、読取り演算Ｒｅａｄ（Ｌ）を実行して、Ｌに収容された識別子のセットを完全に読み取ることができ、アクセス演算Ａｃｃｅｓｓ（Ｌ，Ｑ）を実行して、適切な照会式Ｑを指定することによって、Ｌから識別子を選択することができる。ライブラリＬの読取りは、識別子のランダムなサブセットをＬから取得するプロセスを指し、各識別子は、多数性（またはコピー数）を有する。ＤＮＡシーケンサを使用して、このプロセスを実行することができる。ＤＮＡシークエンシングによる読取り方法は、Ｌのサイズとは独立して、読取り数と呼ばれる、シークエンシング演算が単一の実行でサンプリングおよび報告することができる、Ｌのサブセットの最大サイズ、および読取り長さと呼ばれる、読み取ることができる識別子の最大長さという２つのパラメータによって定義される。特有の識別子が読取りプロセスによってサンプリングされる回数は、その特有のプロセス実行におけるその識別子のコピーカウントまたはカバレージと呼ばれる。本開示では、「アクセス」および「照会」という用語はどちらも、そのセットからの核酸分子のサブセット、データ、または情報を標的とするプロセスに使用されることを理解されたい。本明細書では、これらの用語を区別なく使用することができる。

【０１８４】

$R_{max}$個の識別子の最大のランダム読取り処理量を有する技術を考慮すると、関数≦ｆ（$R_{max}$，δ）は、｜Ｌ｜≦ｆ（$R_{max}$，δ）の場合かつその場合に限り、無視できるほどの障害確率≦１／｜Ｌ｜$^{\delta}$で、その識別子に多数性を有するＬを完全に読み取ることができるように定義することができる。言い換えれば、ライブラリＬは、同じ識別子配列を有する複製識別子分子を有し、したがってｆは、無視できるほどの故障確率で快適に読み取ることができる別個の識別子の数を示す。一例として、現在の技術は、ランダム読取り数$R_{max}$　$25\times10^{6}$および１５０ヌクレオシドの読取り長さに対応する。この技術に対する≦ｆ（$R_{max}$，δ）の適当な推定は、Ｌ内のすべての別個の識別子をサンプリングする際に$10^{-6}$より小さい故障確率を保証するために、$25\times10^{4}$である。

【０１８５】

より小さいライブラリの場合、非ランダム読取りプロセスが実行され、それによって、ハイブリダイゼーションアッセイ、たとえばＤＮＡマイクロアレイ、ＣＲＩＳＰＲに基づくプローブ、またはＰＣＲによって、可能な各識別子の有無が判定される。この非ランダム読取り方法は、たとえばサイズ１，０００またはそれよりも少ない識別子のライブラリを読み取るとき、ＤＮＡシークエンシングより高速かつ安価にすることができる。

【０１８６】

異なる層に対応する成分から識別子が構築されるとき、特有のレベルまたは層に特有の成分を有する識別子にアクセスすることが可能である。照会式は、論理ＡＮＤおよびＯＲによって構成された正規表現として、有向非巡回グラフ（ＤＡＧ）、アクセスプログラムの形で調整される照会木と呼ばれる１組の木構造の形で形式化されたｍａｔｃｈ－ｃｏｍｐｏｎｅｎｔ演算子によって構築される。ＤＡＧは、識別子を収容する様々なコンテナにおいて化学反応が実行される順序を指定する。たとえば、図１３の識別子の組合せ空間における照会は、（０または２）ならびに５および６に一致する成分を有するすべての識別

子のようなものとして指定することができる。そのような表現は、アクセスプログラムとして調整されるとき、この組合せ空間によって画定されるライブラリ内の識別子４および１６にアクセスするはずである。アクセスプログラムは、直列で実行される３つの反応とすることができる。第１の反応では、プローブを使用して、成分０または２を有する識別子を選択する（たとえば、多重化ＰＣＲによる）。第２の反応は、第１の反応からの出力核酸を得て、プローブを使用して、成分５を有する識別子を選択する。最後に、第３の反応は、第２の反応からの出力核酸を得て、プローブを使用して、成分６を有する識別子を選択する。

【０１８７】

Ｌのサブセットにアクセスするために、ｓｅｌｅｃｔｏｒ（ｉ，ｃ$_i$’）が、Ｌ内の識別子のセットを、第ｉの層内の成分Ｃ$_i$の全セットのうちのサブセットｃ$_i$’の部材である第ｉの成分を有する識別子に制限することを可能にする。ｃ$_i$’は、特定の第ｉの成分に対して一致の論理ＯＲを実施する。セレクタｓおよびｔは、ｓおよびｔの両方に一致する識別子にＬを制限するセレクタを取得し、したがって論理ＡＮＤを取得するようにともに構成することができる。論理ＡＮＤは、連続するレベルに適用され、ライブラリＬへの特異性を有するが、図１１および図１３のトライなどのトライＴ（Ｌ）内の１つまたは複数のサブパスにわたって選択を実施する。上記で論じたように、アクセスは、たとえば、ＰＣＲまたは親和性タグ付けを伴う。識別子が、異なる層に対応する成分から構築されるとき、特有のレベルに特有の成分を有する識別子にアクセスすることが可能である。照会式は、論理ＡＮＤおよびＯＲによって構成された正規表現として、有向非巡回グラフ（ＤＡＧ）、アクセスプログラムの形で調整される照会木と呼ばれる１組の木構造の形で形式化されたｍａｔｃｈ－ｃｏｍｐｏｎｅｎｔ演算子によって構築される。ＤＡＧは、識別子を収容する様々なコンテナにおいて化学反応が実行される順序を指定する。

【０１８８】

ｍ個のセレクタのセットは、ｍ個のセレクタすべてのレベルが別個であり、トライＴ（Ｌ）の第１のｍ個のレベルを指すとき、左セレクタまたは５’セレクタと呼ばれる。セレクタのセットσは、σが、Ｔ（Ｌ）のπ個以下のレベルにアクセスする少なくとも１つの左セレクタを収容するとき、アクセサと呼ばれ、ここでπは、アクセス演算に使用されるＰＣＲなどの化学的方法のパラメータである。ＰＣＲの場合、２つのＰＣＲプライマーの各々によって１つの成分を標的とすることによって、２つの成分が反応における標的とされるため、πの値を２とすることができる。アクセサを組み合わせて、Ｔ（Ｌ）の部分木である照会木Ｑを形成する。

【０１８９】

図２０は、例示的な実装形態によるトライＴ（Ｌ）にわたって実行される照会木２０００の図例を示す。トライＴ（Ｌ）は、トライの９個の底部ノードによって示される９個の識別子の組合せ空間を列挙する。図２０に描くように、各識別子は、層１からの１つの成分および層２からの１つの成分という２つの成分を有する。照会Ｑは、グループ２００２ａ内の３つの識別子およびグループ２００２ｂ内の２つの識別子という５つの識別子を選択するように、Ｔ（Ｌ）にわたって実行される。５つの識別子に到達するために５つのパスを構文解析するのではなく、照会Ｑは、低減されたパスセットを伴う。具体的には、照会Ｑは、１つの部分パスおよび２つのフルパスという３つのパスのみのセットを伴う。

【０１９０】

部分パスは、Ｔ（Ｌ）内の下向きのパスであり、根（トライ内の単数の最も上のノード）から始まって内部ノードのうちの１つ（図２０のＴ（Ｌ）の中間レベル内の３つのノードのうちの１つ）へつながる。部分パスは、長さ＜Ｍを有し、複数の層を使用してライブラリＬの識別子を定義する。Ｑ内の部分パスは、２つのノードのみを伴い、したがって部分パスを指定するには、π＝２成分の１つのアクセサで十分である。図２０に描くように、部分パスは、グループ２００２ａ内の３つの識別子を選択する。

【０１９１】

フルパスは、Ｔ（Ｌ）内の根から葉までの下向きのパスであり、根から葉（トライ内の

底部ノードのうちの１つ、固有の識別子配列を示す）まで延びる。フルパスは、長さ＝Ｍを有する。２つのフルパスの各々は、３つのノードを伴い、したがって各フルパスを指定するには、π＝２成分の１つのアクセサおよび１成分の１つのアクセサという２つのアクセサで十分である。図２０の２つのフルパスは、第１のレベルで根ノードおよび別のノードを共用し、したがって２つのフルパスは、サイズπ＝２のアクセサ（根から他のノードへ進む）と、各フルパスの末端に位置する２つの別個の葉に対する２つの別個のセレクタとに分解することができる。

【０１９２】

言い換えれば、照会木ＱがＬ内の識別子のサブセットにアクセスするように実行されるとき、フルパスは、１つの葉（フルパスによって到達される葉）のみを選択し、部分パスは、複数の葉（部分パスのフルパス延長によって到達することができる葉）を選択する。Ｔ（Ｌ）の葉とＬの識別子との間に全単射（１対１のペアリング）が存在し、したがってＬにおけるＱの実行は、照会木Ｑによって到達可能なＴ（Ｌ）の葉によって表される識別子を選択する。$Ｌ_Ｑ$は、Ｑによって選択されたＬのサブセットを示す。図２０の例では、照会木Ｑは、Ｌの５つの識別子を選択し、したがって$Ｌ_Ｑ$は、グループ２００２ａおよび２００２ｂによって示される５つの識別子を収容する。

【０１９３】

Ｑは、アクセス方法によって物理的に可能にされるものより多くの識別子をフェッチすることができるため、照会木は、必ずしも１つの化学反応のみによって実施されるとは限らない。たとえば、所与のＱは、ｆ（$Ｒ_{ｍａｘ}$，δ）より多くの識別子を選択し、またはＱは、π＝２より大きい深さを有し、これは多くのアクセサの実行を必要とする。これらの問題は、アクセスプログラム、すなわちノードが照会木の一部であるＤＡＧによって対処することができ、縁部は、接続されたノード、したがって照会木間の入力／出力を示す。

【０１９４】

Ｌ内の識別子の任意のサブセットＳを、対応する照会木$Ｑ_Ｓ$に変換することができる。Ｓ内の識別子に対応する各葉ｘに対して、$Ｑ_Ｓ$はまず、Ｔ（Ｌ）内の任意のｘにつながるフルパスによって示される木である。$Ｑ_Ｓ$内のノードは、その子（親ノードからの分岐に続くノード）のすべてが$Ｑ_Ｓ$に属する場合、またはそのノードが$Ｑ_Ｓ$の葉である場合、フルとして示される。$Ｑ_Ｓ$は次いで、$Ｑ_Ｓ$からの全ノードのすべての子を剪定し、さらなる剪定が可能でなくなるまでこのステップを繰り返すことによって完成される。残りのノードは、Ｔ（Ｌ）の接続された部分木を形成し、これは厳密にＳの識別子を選択する最も小さい部分木である。この部分木を構成するノードの数は、部分木のサイズを示し、Ｓに対する適当な部分木の最も大きいインスタンスが、Ｓの各識別子に対応するすべての葉ｘに対して長さＭの別個のフルパスを含むため、｜Ｓ｜×Ｍ個のノードによって上限が定められる。剪定の一例として、図２０の照会木Ｑは、２００２ａのすべての３つの識別子がサブセットＳの一部であるため、その子が剪定されたフルノードであると考えられるもので終了する部分パスを有する。

【０１９５】

いくつかの実装形態では、Ｓは、Ｌ内の（ｙ－ｘ＋１）個の識別子隣接範囲Ｒ＝［ｉｄ［ｘ］，ｉｄ［ｙ］］を形成する。Ｒは、Ｔ（Ｌ）の葉の対応する隣接範囲に変換され、$Ｑ_Ｓ$を生成するための上記のステップが適用される。例示的な実装形態によるこの実装形態の一例が、図２１に示されており、図２１は、２分木Ｔ（Ｌ）内のフルパスおよび部分パスのセットにおける識別子（葉によって表される）の隣接範囲Ｒの分解の図例を示す。最も左および最も右のパスは、それぞれ最も左および最も右の識別子ｉｄ［ｘ］およびｉｄ［ｙ］につながる２つのフルパスである。白丸のノードは、木が２進であると仮定して、レベルごとに２つのノードから下降する部分木全体を示す。各レベルにおけるノードの「ファンアウト」（すなわち、成分の数）は、ｃによって示されており、図２１の例で、ｃは２に等しい。Ｒは、（Ｍ－１）個のレベルに対して多くとも２（ｃ－１）個のパスからなる$Ｑ_Ｓ$によって選択され、したがって$Ｑ_Ｒ$は、多くとも２（Ｍ－１）（ｃ－１）個の照会されたパス、したがってセレクタからなる。パスの数は、ｃではなく係数（ｃ－１）

に基づいている。なぜなら、同じ親ノードに由来するｃ個の部分パスがレベルｉに存在する場合、これらの部分パスは、上のレベルの共用の親ノードで終了するが同じ部分範囲に及ぶより短い部分パスに縮小または圧縮することができるからである。

【０１９６】

一般的な照会木Ｑにおいて、アクセスされた識別子が、Ｔ（Ｌ）の対応する葉の順序で順序付けられた場合、識別子は、対応する隣接葉を有する隣接識別子の範囲＜（$s_1$，$t_1$），．．．，（$s_n$，$t_n$）＞を形成し、ここで各（$s_i$，$t_i$）は、識別子／葉の空でない、互いに素である、最大の隣接部分配列である。

【０１９７】

好適な照会木$Q_S$は、（ｉ）照会されたサブセットＳを満たし、（ｉｉ）特定の制限を有する採用された読取り技術に従って実行可能であり、かつ（ｉｉｉ）読取りコストが最小化されたものである。いくつかの実装形態では、必要とされるより多くの識別子を読み取ることが有用であり、したがって$Q_S$は、Ｓ内にないいくつかの偽の正の識別子に到達するが、自動または手動検査を介して後処理ステップで廃棄することができる。照会木$Q_S$に対する実行時間を最小にしたケースは、現在の照会木が及ぶ葉の総数がｆ（$R_{max}$，δ）（所与の読取り方法の単一の実行で読み取ることができる固有の識別子の最大数）より小さくなるまで、木Ｑを最も深いレベルからレベルごとに剪定することによって実施することができる。ノードの数を最小にしたケースは、すべてのノードｕに優先権ｐｒｉｏ（ｕ）が割り当てられる貪欲解によって実現することができ、この解は、現在の照会木からノードの子を除去することによって誘導される利得を測定し、これは、Ｑ内のその子（除去される）の数およびそれらの子の除去によって含まれる余分の葉数の関数として表される。次いで、貪欲解は、剪定された照会木によって含まれる葉の全体的な数がｆ（$R_{max}$，δ）より大きくなるまで、優先権に従ってＱからノードを除去することによって続行される。

【０１９８】

上述したように、たとえば読み取るべき識別子のセットがｆ（$R_{max}$，δ）より大きい場合、または識別子のセット全体のシークエンシングが異常な費用をもたらす場合、アクセスプログラムを使用して、所与の照会の実行可能性に関係する問題を克服することができる。アクセスプログラムＰは、有向非巡回グラフ（ＤＡＧ）であり、（ｉ）各ノードがアクセスまたは読取り演算であり、（ｉｉ）誘導される各縁部が２つのノード間の入力／出力を示し（場合により、標的ライブラリ内に存在する識別子の濃度が増幅される）、（ｉｉｉ）ＤＡＧの根への入力が元の識別子ライブラリＬであり、（ｉｖ）ＤＡＧの出力が、その末端ノードの数と同程度のライブラリＬの（場合により互いに素でない）サブセットからなる。Ｐの実行可能性は、読取り演算が実行されるときはいつでも、読取りへ入力される識別子のプールのサイズがｆ（$R_{max}$，δ）より小さい場合に保証される。実行可能性は、照会木を実行するために使用されるコンテナ内の識別子の濃度に依存し、これは、少なくとも固定の値$m_c$とすることができる。いくつかの実装形態では、濃度要件を満たすために、標的ライブラリＬ内の識別子は、近似定数$m_r$によって複製される。いくつかの実装形態では、ｌ個の長い下向きのパス（根からノード／葉へ）からなる照会木の実行において、ライブラリＬは、ｌ個の別個のコンテナ（たとえば、ウェルプレート内のウェル、試験管）内のアリコートに区画化される（サンプルの体積による区画化）。各アリコートでは、各識別子の最小濃度$m_c$を依然として満たすことができる。等分は、１つのコンテナからｌ個のコンテナへのピペット分注を伴うが、自動化された液体取扱い機械によって自動的に実行することもできる。ライブラリＬのトライＴ（Ｌ）におけるｌ個の長い独立したパスの実行は、１つの識別子当たり$m_c$×ｌ個のコピーを保証する$m_r$＝ｌで複製演算を実行することを伴う。次いで、ｌのアリコート演算を実行して、識別子をｌ個の別個のコンテナに区画化し、各コンテナは、１つの識別子当たり約$m_c$個のコピーを有する。次いで、これらのｌ個のチューブにわたって、ｌ個の長い下向きパスに対してｌ個のアクセサが、個別および／または並列に実行される。

【０１９９】

照会木のパスが比較的短い実装形態では、これらを同じコンテナ上で並列に実行することができ、これは１つの反応または少数の反応のみを必要とする。いくつかのパスが長く、同じサブパスのいくつかを共用する実装形態では、いくつかの反応を同じコンテナ内で併合して、必要とされるアリコートの数ｌを低減させることができる。反応の併合は、照会木の構造に存在する対称性に関係し、したがってアクセスプログラムは、照会木における可能な併合を検出するように構成することができる。

【０２００】

図２２は、トライＴ（３）に適用される照会木Ｑ２２００におけるこの併合プロセスの一例を示し、ここで文字Ａ～Ｉはアクセサを示し、ノードにつながらない分岐は、例示的な実装形態による照会木２２００には存在しないフルトライ２２００のパスである。図２２の例では、照会木２２００は、ＡＤＧ、ＡＤＨ、ＡＦＧ、ＣＤＧ、ＣＤＨ、およびＣＦＧという６つのフルパスを含む。すべてのノードは、そのノードをラベル付けるセレクタまたはアクセサによる化学反応を指定し、反応は、入っている反応（前のノード）から、識別子にわたって実行される。照会木２２００のサイズおよび構造に基づいて、全体で合計１２個のセレクタが使用され、１つのレベル当たり多くとも６つの反応（１つの子ノード当たり１つの反応）が必要とされる。

【０２０１】

図２２の右側の照会ＤＡＧ２２０２は、照会木２２００において共用される接尾辞を併合した結果である。照会ＤＡＧ２２０２は、ＡＦＧとＣＦＧとの間で共用される接尾辞ＦＧ、ＡＤＧとＣＤＧとの間で共用される接尾辞ＤＧ、およびＡＤＨとＣＤＨとの間で共用される接尾辞ＤＨを併合する。その結果得られる照会ＤＡＧ２２０２は、１つのレベル当たり６つのアクセサおよび２つ以下の反応によって実施される。１つのノードにおけるアリコートの数は、そのファンアウトに依存し、したがって照会ＤＡＧ２２０２は、２以下のアリコート演算、１つのレベル当たり多くとも２つのコンテナ、したがって１つのレベル当たり多くとも２つの複製演算によって実施される。

【０２０２】

概して、アクセスプログラムは、ｌ個の下向きパス（セレクタ）からなる照会木を使用して実施される。いくつかの実装形態では、照会木は浅く、これは、パスが長さｐ≦πを有することを意味し、ここでπは、読取り技術に応じて上記で定義したとおりであり、このとき照会木を実行するには１つの反応で十分である。照会木は、パス、すなわちセレクタの結果の論理ＯＲからなり、したがってセレクタは、同じコンテナ上で並列に適用することができる。いくつかの実装形態では、照会木は、πより長いいくつかのパスを有しており、実行において区別される（ｐ／π）個のアクセサのＡＮＤによって実行されなければならない。しかし、照会木が対応する下向きパスで行われる１つまたは複数の繰返しを含む実装形態では、これらの制限を克服することができる。繰返しの形態は、図２１に関連して上述したとおりであり、繰返しは、アクセス演算に対応する照会木内の２つまたはそれよりも多いパスの間で共用される接尾辞である。２つのパスが、図２１の２２００の２つのパスＡＤＧおよびＣＤＧにおけるＤＧの接尾辞などの接尾辞ｓを共用するとき、これらのパスは、｜ｓ｜個のセレクタの最後の配列を共用し、したがって２つのパスによって選択された識別子のセットを同じコンテナ内へ併合することができる。このプロセスは、２２００のような照会木Ｑｓを、２２０２のような照会ＤＡＧ　Ｄｓに変換することを伴い、ここで変換は、Ｑｓにおけるパスの等しい接尾辞を壊すことを伴う。Ｄｓ内のノードは、アクセサ（長いπのセレクタ）によってラベル付けされ、各ノードは、サブパスに従ってコンテナ内で併合された識別子にわたって対応するアクセサを適用することによって実行される反応を示す。アクセスプログラムによって並列に実行されるべき反応の数は、Ｄｓの各レベルにおけるノードの数に等しい。

【０２０３】

いくつかの実装形態では、アクセスプログラムは、その構成パスの論理ＯＲである照会木を有し、論理ＡＮＤにおいてその構成（ｐ／π）アクセサを組み合わせることによって、ｐ≦Ｌの照会パス長さが実行される。アクセスプログラムは、多くとも長さｐのｒ個の

パスの実行を伴い、これらのパスは、各々多くともｒ個のアクセサの（ｐ／π）セットの配列において（１つのパス当たり１つのアクセサ）、レベルごとに分解することができる。そのような各セットは、コンテナにおける１つの反応動作を介して実行される。この実装形態は、たとえばパスがいくつかの接頭辞を共用する（したがってともに実行することができる）場合、またはパスの部分を併合することでアリコートおよび複製演算の数（したがって並列反応の数）を低減させる場合、アクセスプログラムの材料コストを高く推定しすぎる。ｍ個の葉を有する照会木Ｑによって指定されるアクセスプログラムの実行時間は、その深さをπで割った値（１／π）×深さ（Ｑ）に比例する。アクセスプログラムは、照会木Ｑのπ個のレベル当たり１つの反応を実行することによって続行され、これは、場合により別個のコンテナにわたって多くのアクセサを実行することを伴うことができる。実行時間は、複製および等分に要する時間を補償する。いくつかの実装形態では、照会木Ｑは、πの倍数であるレベルでノードを維持し、π成分のサブパスを縮小して、単一のシンボルであると考えられる長さπのメタ成分を形成することによって縮小される。すべてのπ個のレベルにおいて、ｃ$^{\pi}$個のパスが調査され、したがって縮小されたトライは、Ｌ／πの深さおよびｃ$^{\pi}$のファンアウトを有する。１つのノードは、ファンアウトｆを有し、このとき親ノードからのその入力は、ｆ回複製される。木Ｑから導出された照会ＤＡＧ　Ｄに従って照会が実行される実装形態では、実行時間は、（１／π）×深さ（Ｄ）に、各ノードのファンアウトを考慮してＤのノードに基づいて事前に判定される複製時間を足した値によって近似される。上記で論じた照会ＤＡＧの定義によれば、深さ（Ｄ）＝深さ（Ｑ）であり、かつ｜Ｄ｜≦｜Ｑ｜であるため、この実行時間は、Ｑに対する実行時間より小さくすることができる。

【０２０４】

アクセサがπ個の成分によって形成されると仮定すると、アクセスプログラムの設定時間は、Ｑのサイズをπで割った値に比例する。設定時間は、サイズ｜Ｑ｜、またはｍ個の個々のパスの長さの和をπによって割った値（１／π）×ｍ×深さ（Ｑ）のどちらかより小さいほうによって、上限が制限される。Ｑ$_{S}$に対応する照会ＤＡＧ　Ｄ$_{S}$が、設定時間を指定するために、Ｑの代わりに使用される。

【０２０５】

本明細書に論じる読取りおよびアクセス演算は、ＤＮＡに基づく記憶システムで他の演算を実行するための基本を形成する。

【０２０６】

核酸配列に記憶されたデータのランクおよびフェッチ演算

【０２０７】

図２３および図２４は、例示的な実装形態によるランク演算を実行して、ストリング内の特定の位置を含むその特定の位置までのデータストリングの範囲に対するシンボルまたはビット値のカウントを判定する例示的な方法を描く。ランク演算は、上述した複数のアクセスおよび読取り演算を含む。同様に、フェッチ演算を実行して、シンボルストリングのサブセットを取り出すことができる。上記で論じたように、本開示の態様は、２進ストリングの記憶、索引付け、および探索に関する。長さｎの２進ビットのストリングＳは、サイズｃのセット内で選択されたＭ個の成分によって構築された識別子のプールＬ$_{S}$を介して識別子核酸分子によって表すことができる。たとえば、組合せライブラリＬのサイズ｜Ｌ［Ｍ］｜は、ｃ$^{L}$≧ｎに等しく設定され、したがって層の数Ｍは、ｌｏｇ$_{c}$ｎによって近似される。ストリングＳは、識別子のプールとして符号化され、したがってＳ内の位置ｘのビットＳ［ｘ］がビット値１を有する場合、位置ｘの識別子ｉｄ［ｘ］はＬ$_{S}$に属する。いくつかの実装形態では、これは、ビット値１に対応するＬ内の識別子のみが構築され、後にプールされ、ビット値０に対応するＬ内の識別子は構築されないことを意味する。

【０２０８】

本開示は、ランク演算ｒａｎｋ（ｘ）を実行するシステムおよび方法を含み、ランク演算は、２進ストリングに対して、接頭辞ストリングＳ［０，ｘ］（ビットストリングＳの

うち位置ｘに先行して位置ｘを含む部分）内の１の数のカウントを返す。非２進ストリングの場合、ランク演算は、接頭辞ストリング内の特有のシンボル値を有するシンボルの数のカウントを返す。本明細書に記載する例のいくつかは２進ストリングに関するが、本開示は、３つ以上の可能なシンボル値を有するシンボルストリングに対するランク演算にも適用されることが理解されよう。

【０２０９】

いくつかの実装形態では、ｂは、２進ビットストリングＳの長さを示す整数ｎを符号化するために必要とされるビットの数である。ビットの数ｂは、$\log_2(n+1)$によって推定される。ストリングＳは、サイズｂのブロックに分けられ、ストリングＳ内に位置ｘを含むブロックは、Ｂ（ｘ）によって示すことができる。ストリングＳの一方の末端で０から計数すると、順序付けられた索引Ｂ（ｘ）は、ｆｌｏｏｒ（ｘ／ｂ）に等しい。ランクの算出は、（ｉ）Ｓ［０，ｂ×Ｂ（ｘ）－１］内の１の数（ｘを収容するブロックＢ（ｘ）に先行するブロック内の１の数）を計数すること、および（ｉｉ）Ｓ［ｂ×Ｂ（ｘ），ｘ］内の１の数（ｘを含むｘまでのブロックＢ（ｘ）内の１の数）を計数することという２つのステップに分解することができる。これらのステップは、任意の順序で行うことができる。次いで、両方のステップからのカウントを合計して、位置ｘにおけるランクを判定する。いくつかの実装形態では、アクセス演算および読取り演算の各対が別個の標的ライブラリに適用され、別個の標的ライブラリが各ステップ（ｉ）および（ｉｉ）に対応するため、演算ｒａｎｋ（ｘ）は、並列に実行可能な２つのアクセス演算および２つの読取り演算からなるアクセスプログラムを介して実施される。

【０２１０】

本明細書に記載する例のいくつかは、上記の２つのステップに関するが、本開示は、同等の結果に到達するランクを判定する他の方法も含む。たとえば、いくつかの実装形態では、上述した２つのステップがわずかに修正される。具体的には、ランクの算出は、（ｉ）Ｓ［０，（ｂ＋１）×Ｂ（ｘ）－１］内の１の数（ｘを収容するブロックＢ（ｘ）に先行してブロックＢ（ｘ）を含むブロック内の１の数）を計数すること、および（ｉｉ）Ｓ［ｘ＋１，（ｂ＋１）×Ｂ（ｘ）］内の１の数（ステップ（ｉ）内に含まれていたがｘ後にのみ行われるブロックＢ（ｘ）内の１の数）を計数することという２つの異なるステップに分解することができる。次いで、ステップ（ｉｉ）からのカウントをステップ（ｉ）のカウントから引いて、ステップ（ｉ）および（ｉｉ）で２重に計数された１の数を除去することによって、位置ｘのランクを取得する。より概略的には、ストリングＳは、ブロックサイズｗに分けることができ、ｗはｂと同じである必要はない。いくつかの実装形態では、アクセス演算および読取り演算の各対が別個の標的ライブラリに適用され、別個の標的ライブラリが各ステップ（ｉ）および（ｉｉ）に対応するため、演算ｒａｎｋ（ｘ）は、並列に実行可能な２つのアクセス演算および２つの読取り演算からなるアクセスプログラムを介して実施される。

【０２１１】

図２３は、例示的な実装形態による本明細書に記載する方法による識別子を使用して記憶されたｎ＝３０ビットの２進ストリングＳ内のｒａｎｋ（２２）の計算のための一例を示す。したがって、この演算は、Ｓ内の位置ｘ＝２２を含む位置ｘ＝２２までの位置における、ビット値１の数を判定する。この例では、標的ライブラリ$L_S$は、Ｓ内の値１に設定された１３ビットに対応する１３個の識別子を含む。識別子は、各々３つの成分から構成され、３つの成分は、図２３の各トライ内の３つのレベルに対応する。したがって、図２３のストリング$S$の上のトライ$T_S$「３」は、１３個の葉（１のビットに対する１３個の識別子に対応する）、これらの葉の上の３つのレベル、およびｃ＝４のファンアウトからなる。既存の葉（ビット値１に対応する識別子）につながるパスのみが描かれている。

【０２１２】

図２３の第１の照会木は、上記で論じたランク演算のステップ（ｉｉ）に対応する。この例では、ブロックサイズｂは、$\log(n+1)=5$に等しく、位置２２を収容するブロックは、Ｂ（２２）＝４である。照会木$Q_R$（木構造においてより太い線で示される）

は、位置２０から位置２２の範囲Ｒ［２０，２２］に及ぶ。$Q_R$は、それぞれビット値１、１、および０に対応する位置２０、２１、および２２で３つの葉を選択する３つのフルパスからなる。これらの値のうちの２つのみが１に等しく、したがってそれぞれ位置２０および２１における識別子ｉｄ［２０］およびｉｄ［２１］が取り出される。この照会木$Q_R$に対するカウントは、上述したステップ（ｉｉ）に従って、位置ｘ＝２２を含み位置ｘ＝２２までのＢ（２２）内の１の数に対応する$n_S=2$に設定される。

【０２１３】

次に、ランク演算のステップ（ｉ）に従って、Ｂ（２２）に先行するブロック内の残りの１を判定することができる。このステップは、各々ｂ＝５ビットのｃｅｉｌｉｎｇ（ｎ／ｂ）＝６個のカウンタを符号化する２進カウンタストリングＣを参照することによって実行される。Ｃは、本明細書に記載する方法による識別子を使用して記憶され、したがってＳと同様にアクセスして読み取ることができる。Ｃの各５ビットカウンタは、各カウンタに先行するＳの接頭辞内の１の数を計数する。Ｓは、参照のためにＣの下でコピーされる。第１のカウンタに先行するＳの接頭辞がないため、Ｃの第１の５ビットカウンタは、２進で０に等しい。第２の５ビットカウンタは、３つの１を収容するＳの第１のブロックに続くため、２進で３に等しい「００００１１」である。Ｃのビットは、Ｃに適用される照会木$Q_C$のｃ＝４ファンアウトに対応する４ビットのグループを示すために、垂直の点線によって分離される。Ｃに適用される$Q_C$は、Ｂ（２２）の接頭辞Ｓ［０，１９］内の１ビットの数のカウンタに対応する範囲Ｃ［２０，２４］内の識別子を選択する。$Q_C$は、ノードｕで終了する１つの部分パス、および位置２４のビットに及ぶ１つのフルパスからなる。対応するビットが０に設定されているため、位置２４における識別子ｉｄ［２４］は存在せず、したがってフル照会パスは何も取り出さない。ノードｕで終了する部分パスは、それぞれ位置２２および２３における２つの識別子ｉｄ［２２］およびｉｄ［２３］を取り出す。カウンタＣ［２０，２４］のビットは、Ｂ（２２）の接頭辞Ｓ［０，１９］内の１の数に対応する整数値６を符号化する「００１１０」を読み出す。この照会木$Q_C$に対するカウントは、上述したステップ（ｉ）に従って、Ｂ（２２）までのＳ内の１の数に対応する$n_C=6$に設定される。アクセスプログラムを介したこの例示的なランク演算の実行は、$n_S+n_C=2+6=8$を返し、これはＳ［０，２２］内の１ビットの正しい数である。

【０２１４】

図２３の例によって上述および図示したように、ランクの算出は、（ｉ）Ｓ［０，ｂ×Ｂ（ｘ）－１］内の１の数（ｘを収容するブロックＢ（ｘ）に先行するブロック内の１の数）を計数すること、および（ｉｉ）Ｓ［ｂ×Ｂ（ｘ），ｘ］内の１の数（ｘを含むｘまでのブロックＢ（ｘ）内の１の数）を加算することという２つのステップに分解することができる。この演算は、長さｎのビットストリングＳに対して形式化されており、次の段階に従って実行される。段階１は、ブロックＢ（ｘ）の接頭辞Ｓ［０，ｂ×Ｂ（ｘ）－１］内の１の数$n_1$（Ｃ）を算出することを伴う。接頭辞は、特定のビットまたはブロックに先行するビットを指す。第１の標的ライブラリ$L_C$は、ストリングＳに対するカウンタストリングである２進ストリングＣ［１，ｂ×Ｂ（ｎ）］を表す。カウンタストリングＣは、Ｃ［ｉｂ，（ｉ＋１）ｂ－１］がＳの値ｒａｎｋ（ｉｂ－１）のｂビット内に２進表現を記憶するように定義され、ここでｉ＝１、．．．Ｂ（ｎ）－１である。Ｃは、長さ≦ｎを有する（Ｓの長さより小さくまたはそれに等しい）。Ｃ（０）は、定義によれば０に等しいため、記憶されない。段階１は、指定の位置ｘを収容するブロックＢ（ｘ）までのＳのランクを符号化するＣのカウンタに対応する、範囲$R_C$＝（ｉｄ［ｂ×Ｂ（ｘ）］，ｉｄ［ｂ×（Ｂ（ｘ）＋１）－１］）内で第１のライブラリ$L_C$の識別子をフェッチする照会木$Q_C$を作成することを伴う。$Q_C$は、約$\log_c n$の深さおよび約ｃＬのサイズ、すなわち深さにファンアウトを掛けた値を有し、ここでサイズは、ブロックサイズｂに依存しない。次いで段階１は、Ｙ＝ａｃｃｅｓｓ（$L_C$，$R_C$）（範囲$R_C$内の$L_C$の識別子にアクセスする）、次いでｒｅａｄ（Ｙ）という２つの演算を介して、$R_C$によって境界が定められたＣのサブストリングを構成するビットを読み取るアクセスプログラムを作成す

ることを伴い、ここでＹは、アクセス演算を介して取得される識別子のサブセットである。このアクセスプログラムは、多くともｂ個の識別子の範囲にわたって読取り演算が実行されるため、上記で論じた基準に従って実行可能であり、ここでｂは約ｌｏｇ（ｎ）であり、これはｆ（$R_{max}$，δ）より小さい。アクセスプログラムによって取り出される識別子は、$R_C$によって境界が定められたＣのサブストリング内の１ビットの位置を示す。段階１は、それらの位置のみが１に設定されたｂビットの２進サブストリングによって符号化された整数値に、$n_1$（Ｃ）を設定することによって終わる。

【０２１５】

段階２は、Ｓ［ｂ×Ｂ（ｘ），ｘ］内の１の数$n_1$（Ｒ）を算出することを伴い、Ｓのサブストリングは、位置ｘを収容するブロックＢ（ｘ）に対応する。第２の標的ライブラリ$L_S$は、２進ストリングＳを表し、したがってＳ内の対応する位置のビット値が１に等しい場合、識別子が物理的に存在し、すなわちＳ［ｘ］＝１の場合かつその場合に限り、ｉｄ［ｘ］が存在する。段階１は、位置ｘを含む位置ｘまでのＳ内のブロックＢ（ｘ）の位置に対応する、範囲$R_S$＝（ｉｄ［ｂ×Ｂ（ｘ）］，ｉｄ［ｘ］）内の第２のライブラリ$L_S$の識別子をフェッチする照会木$Q_R$を作成することを伴う。$Q_R$は、約$\log_c n$の深さおよび約ｃＬのサイズ、すなわち深さにファンアウトを掛けた値を有する。次いで段階１はＸ＝ａｃｃｅｓｓ（$L_S$，$R_S$）（範囲$R_S$内の$L_S$の識別子にアクセスする）、次いでｒｅａｄ（Ｘ）という２つの演算を介して、$R_S$によって境界が定められたＳのサブストリングを構成するビットを読み取るアクセスプログラムを作成することを伴い、ここでＸは、アクセス演算を介して取得される識別子のサブセットである。アクセスプログラムによって取り出される識別子は、$R_S$によって境界が定められたＳのサブストリング内の１ビットの位置を示す。段階２は、取り出された識別子の数に等しい$n_1$（Ｒ）を設定することによって終わる。ランク演算は、それぞれ段階１および２から値$n_1$（Ｃ）＋$n_1$（Ｒ）を返すことによって完了し、ここでランクは、０から位置ｘの範囲におけるＳ内の１ビットの数である。

【０２１６】

図２４は、例示的な実装形態によるビットストリング上でランク演算を実行する流れ図２４００を示す。流れ図２４００の方法では、図２３に関して上述した方法ならびに段階１および２による概略的なケースを使用することができる。ステップ２４０２で、上記のＳなどのビットストリングを表す識別子の第１のプールが取得される。ステップ２４０４で、上記のＣなどのカウンタシンボルストリングを表す識別子の第２のプールが取得される。ステップ２４０６で、特定の値のビットのランニングカウントを示す識別子を標的とするように、連続することができる第２の一連のプローブを使用して第２のプールにアクセスすることによって、第１のカウントが取得される。所与の値のビットの数のランニングカウントを示すカウンタシンボルを表す標的とされた識別子は、（１）特定のビットに先行するｗ個のビットのすべてのブロック、または（２）特定のビットを含むｗ個のビットのブロックを含む、特定のビットに先行するｗ個のビットのすべてのブロックのいずれかに対して取得される。ステップ２４０８で、識別子を標的とするように、連続する第１の一連のプローブを使用して第１のプールにアクセスすることによって、第２のカウントが取得される。標的とされた識別子は、（１）特定のビットに先行しもしくは特定のビットを含む、ステップ２４０６で計数されていないビットを表す、または（２）ステップ２４０６で計数されたが特定のビットに先行しないもしくは特定のビットを含まないビットを表す。ステップ２４１０で、第１のカウントおよび第２のカウントから、ビットストリング内の特定のビットのランクが取得される（たとえば、カウントの和または差を判定することによる）。

【０２１７】

ビットストリング内の各ビットは、ビット値およびビット位置を有する。各プールは、固体、液体、または固体の形態を有することができ、複数の識別子核酸分子を形成することによって形成される。各識別子は、それぞれのビット位置に対応し、Ｍ個の選択された成分核酸分子を物理的に組み立てることによって形成される。Ｍ個の選択された成分の各

々は、Ｍ個の異なる層に分離された別個の成分核酸分子のセットから選択される。識別子は、ビットストリングを表すように、第１のプール内に収集することができ、したがってビット値は、第１のプール内の対応する識別子の有無によって示される。同様に、識別子の第２のプールは、各々カウンタストリング内のビットを表す識別子を収集することによって形成される。いくつかの実装形態では、第１のプールは第２のプールと同じであり、他の実装形態では、第１のプールおよび第２のプールは別個である。

【０２１８】

いくつかの実装形態では、第１のプール内の対応する識別子の物理的な存在は、ビット値１を示し、対応する識別子の物理的な不在は、ビット値０を示す。各カウンタシンボルは、ｂ個のカウンタビットのストリングによって表すことができ、ｂは、関数$\log_2$（ｎ＋１）の上限によって判定することができ、ここでｎはビットストリングの長さである。ｂ個のカウンタビットの各ストリングは、特有の値、たとえば１または０を有するビットストリング内のすべてのｗ個のビットに対するビットの数のランニングカウントを表すことができる。カウンタシンボルストリングは、ｎをｗで割った上限個のカウンタシンボルを含むことができ、長さｎを有するカウンタビットのストリングによって表される。いくつかの実装形態では、初期カウンタシンボルは、値０を有し、これは、すべて値０を有するｂ個のカウンタビットのストリングによって表される。特定のビットが、ｗ個のビットの第１のブロック内にある場合、ｗ個のビットの第１のブロックに先行するランニングカウントは０である。

【０２１９】

第１および第２のカウントは、本明細書に記載する読取り演算に従って、各照会から標的とされた識別子を読み取ることによって取得される。たとえば、第１のカウントは、２４０６で標的とされた識別子に対応するカウンタシンボル値を読み取ることによって取得され、または第２のカウントは、２４０８で標的とされた識別子を読み取ることによって取得される。いくつかの実装形態では、各ストリングの第１の索引は、０に等しく設定される。ステップ２４０８で第１のカウントを取得するために使用されたカウンタシンボルは、範囲０～ｗ×Ｂ（ｘ）－１内で値１を有するビットストリング内のビットの数に対応することができ、ここでｘはビットストリング内の特定のビットの位置に対応し、Ｂ（ｘ）はｘをｗで割った下限である。第２のプールから少なくともｂ個の識別子を標的とすることができ、標的の少なくともｂ個の識別子は、範囲ｂ×Ｂ（ｘ）～ｂ×（Ｂ（ｘ）＋１）－１内とすることができる。

【０２２０】

いくつかの実装形態では、第２のカウントは、範囲ｗ×Ｂ（ｘ）～ｘ内で値１を有するビットストリング内のビットの数に対応し、ここでｘはビットストリング内の特定のビットの位置に対応し、Ｂ（ｘ）はｘをｗで割った下限である。ステップ２４０８で、特定のビットを含むｗ個のビットのブロックに対応するカウンタシンボルを表す第２のプールから照会された識別子を標的とすることによって、第１のカウントを取得することができ、ステップ２４１２で、２４０８で計数されたが特定のビットに先行しないまたは特定のビットを含まないビットを表す２４１０で標的とされた固有の識別子を標的として計数することによって、第２のカウントが取得される。したがって、２４１４で、第１のカウントから第２のカウントを引くことによって、ランクが取得される。

【０２２１】

いくつかの実装形態では、カウンタストリングブロックサイズｗを、ビットストリングブロックサイズｂに等しく設定することができる。別法として、カウンタストリングブロックサイズｗを、１に設定することができる。２４０８の第１のカウントは、特定のビットを含むｗ個のビットのブロックに対応するカウンタシンボルを表す２４０６で第２のプールで照会された識別子を標的とすることによって取得することができ、ランクは第１のカウントと同等である。したがって、ステップ２４１２および２４１４をこの方法から省略することができる。

【０２２２】

いくつかの実装形態では、識別子核酸の第１のプールは、ビットストリングの変換を表し、したがって識別子の有無は、ビットストリング内のいずれのビット値とも直接相関しないが、コードワードと呼ばれる隣接して順序付けられた識別子のブロックは、ビットストリング内のビットのブロックに変換させることができる。コードワードは、固定の数の可能な固有の識別子核酸分子からの、固定の数の固有の識別子核酸分子の存在を含む。追加の情報を使用して、第１および第２のプールからの識別子核酸分子の書込み、アクセス、および読取りのエラーを検出および補正することができる。前記追加の情報は、第１および第２のプールの識別子に記憶される。

【０２２３】

いくつかの実装形態では、２４０６の第１のカウントは、特定のビットに先行するｗ個のビットのすべてのブロックを表し、２４０８の第１の一連のプローブは、特定のビットに先行しまたは特定のビットを含む、２４０６で計数されていないビットを表す第１のプール内の１つまたは複数の別個の識別子核酸分子を標的とし、２４１０で第１および第２のカウントを合計することによって、ビットストリング内の特定のビットのランクが取得される。他の実装形態では、２４０６の第１のカウントは、特定のビットに先行し、特定のビットを含むｗ個のビットのブロックを含む、ｗ個のビットのすべてのブロックを表し、第１の一連のプローブは、２４０６で計数されたが特定のビットに先行しないまたは特定のビットを含まないビットを表す第１のプール内の１つまたは複数の別個の識別子核酸分子を標的とし、２４１０で第１のカウントから第２のカウントを引くことによって、ビットストリング内の特定のビットのランクが取得される。

【０２２４】

いくつかの実装形態では、Ｓ内で読み取るべき２進ブロックのサイズが調整される。ブロックサイズを増大させることで、Ｃ内のカウンタの数を低減させ、したがってＣの長さを低減させる。たとえば、ｗをＳ内の２進ブロックの長さとして設定することによって、Ｃの長さは約（ｎ／ｗ）ｌｏｇ（ｎ）になり、ｗは段階２でｒｅａｄ（Ｘ）からフェッチされたビットの量である。上記の分解では、ｗをｌｏｇ（ｎ）にほぼ等しく設定することによって、Ｃのサイズは、Ｓのサイズｎより小さくまたはそれに等しくなるように選択された。

【０２２５】

方法２４００によって説明したランク演算は、２進ビットストリングに注目しているが、方法２４００はまた、シンボルストリングにも適用することができることを理解されたい。いくつかの実装形態では、ビットストリングは、シンボルストリングを表し、ランクは、シンボルストリング内の特定のシンボルに対して取得される。シンボルストリング内のシンボルは、シンボル値のセットから選択され、２４０４のカウンタシンボルストリングは、特定のシンボル値を有するシンボルの数のランニングカウントを示す。いくつかの実装形態では、２４０４で識別子の複数の第２のプールが取得され、各第２のプールは、特有のシンボル値のインスタンスの数を計数するカウンタシンボルの異なるストリングを表し、カウンタシンボルの異なる各ストリングが、特有のシンボル値に対応するインスタンスを計数する。

【０２２６】

図２３の上記の例では、別法として、Ｓ内の０ビットの数を計数することによって、任意の２進ランク演算を実行することができ、この演算は、$rank_0(x)$によって示される。$rank_0(x)$は、$rank_0(x) = x - rank_1(x) + 1$として算出することができる。シンボル値のセットから選択されたシンボルストリング上でランク演算を実行して、シンボルストリング内の特定のシンボルに対するランクを取得することもできる。たとえば、カウンタシンボルストリングは、特有のシンボル値を有するシンボルの数のランニングカウントを示す。ステップ２４０４で、異なる識別子プールは、特有のシンボル値のインスタンスの数を計数するカウンタシンボルの異なるストリングを表し、カウンタシンボルの異なる各ストリングが、対応する特有のシンボル値のインスタンスを計数する。

【０２２７】
いくつかの実装形態では、ランク演算が大きいビットストリング（たとえば、大きい入力メッセージ）上で実行される場合でも、所与のランク演算を実行するために読み取らなければならない識別子の数は非常に小さい。たとえば、テラビットサイズのメッセージ内の特定のビットのランク付けは、構成に応じて、わずか４０～１００個の識別子を読み取ることを伴う。この少量の識別子によって、より高い関連コストを有するＤＮＡシークエンシングのようなより高処理量の方法を使用するのではなく、ＤＮＡマイクロアレイまたはＰＣＲ（たとえば、ｑＰＣＲまたはデジタルＰＣＲ）などの低処理量のハイブリダイゼーション方法によって、読取りを実行することができる。
【０２２８】
本明細書に記載するＤＮＡに基づく記憶システム上で実行することができる別の演算はフェッチ演算であり、これはたとえば、整数アレイを取り扱うためにカウンタアレイＣから１つのエントリを取り出す上記のプロセスを拡大することによって行われる。各々ｂビットからなるｎ個の整数のアレイを考慮すると、アレイは、固定サイズｂの表現でｎ個の整数を直列化することによって、２進ストリングに変換することができる。変換された２進ストリングは、直列化が長さｎ×ｂの２進ストリングをもたらすため、Ａ［０，ｎｂ－１］として表される。Ａは、本明細書に記載する方法によって符号化された識別子プール$L_A$によって表され、ここで各識別子は、Ａで１のビット値に対応する場合、物理的に構築される（すなわち、Ａ［ｘ］＝１の場合かつその場合に限り、ｉｄ［ｘ］）。フェッチ演算は、範囲［ｘ，ｙ］内に生じる整数を返すｆｅｔｃｈ（ｘ，ｙ）として定義され、範囲およびストリング長さによって判定された位置（すなわち、ｉ＝ｘ、．．．．、ｙの場合は位置ｉ×ｂ）で始まる長さｂの２進ストリングによって表される。フェッチ演算は、第１の位置ｘｂおよび最後の位置（ｙ＋１）ｂ－１によって境界が定められたＡのサブアレイを表す$L_A$の識別子にアクセスする。フェッチ演算は、１つのアクセス演算および１つの読取り演算からなるアクセスプログラムによって実施される。
【０２２９】
フェッチ演算のためのアクセスプログラムは、次のように実行することができる。標的ライブラリ$L_A$は、上述した２進ストリングＡを表す。上述したように、第１の位置ｘｂおよび最後の位置（ｙ＋１）ｂ－１によって境界が定められた範囲$R_A$（すなわち、$R_A$＝（ｉｄ［ｘ×ｂ］，ｉｄ［（ｙ＋１）ｂ－１］））内の識別子を選択する照会木$Q_A$が作成される。範囲$R_A$は、範囲ｘ～ｙの位置（すなわち、［ｘ，ｙ］）で生じる入力アレイ（直列化されていない整数アレイ）の整数を表すＡのビットをちょうどすべて含む。本明細書に記載するアクセス方法によって、$R_A$によって境界が定められたＡのサブストリングがアクセスされ、本明細書に記載する読取り方法によって、そのサブストリング内に含まれる識別子が読み取られる。このアクセスプログラムは、$\log_c nb$に等しい約$\log_c A$の深さを有し、$\log_c nb$に等しい約$\log_c A$のサイズを有する。範囲$R_A$は、（ｙ－ｘ＋１）×ｂ個のビットに等しいサイズを有する。アクセスプログラムによって取り出される識別子を使用して、入力整数アレイの範囲［ｘ，ｙ］内の位置で記憶されたｙ－ｘ＋１個の整数を再構築することができる。このフェッチ演算は、約（ｙ－ｘ）の実行時間で実行することができる。
【０２３０】
図２５は、例示的な実装形態によるシンボルストリング上でのフェッチ演算の実行のための流れ図を示す。上述した方法および論理を図２５のこの方法で使用して、フェッチ演算を実行することができる。ステップ２５０２は、シンボルストリングを表す識別子の第１のプールを取得することを伴う。ステップ２５０４は、第１の一連のプローブを有する第１のプールにアクセスして、第１のプールからの識別子のサブセットを有する第２のプールを作成することを伴い、各プローブは、第１のプールの識別子の成分を標的とする。ステップ２５０６は、第２のプール内の識別子のサブセットの配列を読み取ることを伴う。ステップ２５０８は、読み取った配列を使用して、シンボルストリングのサブセットを取得することを伴う。

【０２３１】
シンボルストリングでは、各シンボルが、シンボル値およびシンボル位置を有する。いくつかの実装形態では、ステップ２５０２は、複数の識別子核酸分子を形成することを伴い、各識別子がそれぞれのシンボル位置に対応し、Ｍ個の選択された成分核酸分子を物理的に組み立てることによって形成される。Ｍ個の選択された成分の各々は、Ｍ個の異なる層に分離された別個の成分のセットから選択される。各層の成分は、図１３に関連して説明したように、論理的に順序付けることができる。識別子は、シンボルストリングを表すようにプール内に収集することができ、したがってシンボル値は、前記プール内の対応する識別子の有無によって示される。
【０２３２】
いくつかの実装形態では、ステップ２５０４で、第１の一連のプローブは連続しており、成分のＭ個の層を表す照会木内の下向きの部分パスまたはフルパスに対応する。下向きのフルパスは、Ｍ個のプローブを含む根から葉までのパスに対応し、したがって連続する一連のＭ個のプローブは、単一の識別子を標的とする。下向きの部分パスは、Ｍ個未満のプローブに対応し、したがって連続する一連のプローブは、異なる配列を有する識別子の複数の母集団を標的とする。異なる配列を有する識別子の複数の母集団は、少なくとも第Ｍの層内の異なる成分に対応することができる。
【０２３３】
いくつかの実装形態では、ステップ２５０４で、第１のプールを照会することは、第１の連続する一連のプローブのうちの第１のプローブを使用して、第１のプール内の第１の層に第１の成分核酸分子を捕捉することと、第１のプールを別個のコンパートメント内の少なくとも２つのプールに分離することと、追加のプローブを使用して、少なくとも２つのプール内の第２の層に成分核酸分子を捕捉することとを伴う。第１の連続する一連のプローブは、サブセットを含むシンボルストリングの望ましい部分を取得するように設計することができる。望ましい部分は、ステップ２５０４で取得したサブセットに密接に対応することができる。
【０２３４】
いくつかの実装形態では、ステップ２５０２で取得した第１のプールは、少なくとも２つの複製プールに分割され、ステップ２５０４～２５０８は、前記複製プールの各々で実行される。第１のプールは、少なくとも２つの複製プールに分割される前に複製することができる（たとえば、ＰＣＲを介して）。ステップ２５０４のプローブは、ＰＣＲプライマーとすることができ、アクセスはＰＣＲによって実行される。別法として、ステップ２５０４のプローブは、親和性タグ付きオリゴヌクレオチドとすることができ、アクセスは親和性プルダウンアッセイによって実行される。いくつかの実装形態では、方法２５００は、プローブのサブシリーズによって、識別子の第１のプールにアクセスし、識別子の中間プールを作成することをさらに含む。中間プールは、少なくとも２つの複製プールに分割することができる。プローブの後続のサブシリーズによって第１の中間プールにアクセスして、第２の中間プールまたは第２のプールを形成することができる。少なくとも２つの中間プールを組み合わせて、別の中間プールを形成することができる。
【０２３５】
いくつかの実装形態では、フェッチ演算は、識別子の読取りのためにＤＮＡマイクロアレイまたはＰＣＲ（たとえば、ｑＰＣＲまたはデジタルＰＣＲ）のようなハイブリダイゼーションに基づく手法を使用する。たとえば、標的とされた識別子が少量である場合、低処理量の方法を使用するとより費用効果を高くすることができるが、識別子の大きいサブセットが読み取られる場合、高処理量の読取りのために、ＤＮＡシークエンシングを使用することができる。
【０２３６】
核酸分子に記憶されたデータのパターン探索
【０２３７】
上述したように、ＤＮＡに基づくプラットホーム上で実行することができる別の演算は

、ストリングのＢＷＴに依拠するカウント演算およびランク演算である。カウント演算の基本は、並列に実行することができる２つのランク演算を使用することである。なぜなら、各ランク演算がＢＷＴおよびカウンタアレイの異なる部分を要求し、またはこれらの部分が重複し、それらの演算の結果を再利用するようにそれらの実行を構築することができるからである。ＢＷＴ、カウンタアレイ、およびランク演算を導入する探索方法について、以下に説明する。

【０２３８】

データ構造は本明細書に記載するシステムで実施可能であり、カウント演算を実行して、ストリングにおけるパターンＰの発生の数を計数することができる。このカウント演算について、図２６～図３０に関連して説明する。カウント演算の基本は、並列に実行することができる２つのランク演算を使用することである。なぜなら、各ランク演算がＢＷＴおよびカウンタアレイの異なる部分を要求し、またはこれらの部分が重複し、それらの演算の結果を再利用するようにそれらの実行を構築することができるからである。パターン発生の取出しは、必ずしも２進探索を介して実施されないが、本明細書に記載するシステムは、ＢＷＴ、カウンタアレイ、およびランク演算を導入する探索方法を使用することができる。これらの演算／データ構造は、圧縮された形で記憶しながら、それでもなおエントリの取出しを可能にすることができる。

【０２３９】

カウント演算は、たとえばメッセージのＢｕｒｒｏｗｓ－Ｗｈｅｅｌｅｒ変換行列の最後の列であるビットストリングＬを表す識別子の第１のプールを取得することによって実行することができる。各識別子は、本明細書に記載する方法によって組み立てられており、１つまたは複数のプローブに個々に結合することが可能である。ビットストリングＬから導出されたカウンタシンボルストリングを表す識別子の第２のプールが取得され、これは特有のビット値を有するビットの数のランニングカウントを表す。メッセージ内の特定のビットパターンのカウントは、第１および第２のプールからの識別子に選択的にアクセスすることによって取得される。一連のプローブを使用して、第１および第２のプールからの識別子にアクセスする。ランニングカウントを使用して、たとえば後述するＬＦマッピングを使用することによって、ＢＷＴ行列の第１の列Ｆを再構築することができる。

【０２４０】

図２６は、例示的な実装形態による２進ビットストリングＳにおける特定のパターンＰの発生の数を計数するカウント演算方法２６００のための流れ図を示し、２進ビットストリングＳは、たとえば、メッセージまたは他の形態の情報とすることができる。ＳのＢＷＴ　Ｌは、第１のプール内の識別子によって表現を介して記憶される。ステップ２６０２は、ＢＷＴ　Ｌを表す識別子の第１のプールを取得することを伴う。ステップ２６０４で、識別子の第２のプールが取得され、第２のプールの識別子は、Ｌから導出されたカウンタシンボルストリングを表す。ステップ２６０６で、本明細書に記載する方法を使用して第２のプールの識別子にアクセスして、Ｌ内のビット値１の発生の総数を示すカウンタシンボルを表す識別子を取り出す。ステップ２６０８は、取り出した識別子からカウンタシンボル値を読み取って、Ｌ内の各ビット値の発生の総数（たとえば、２進ストリングに対する１または０の総数）を計数することを伴う。ステップ２６１０で、ステップ２６０８で計数した総数を使用して、Ｂｕｒｒｏｗｓ－Ｗｈｅｅｌｅｒ行列の第１の列Ｆが構築される。ステップ２６１２で、パターンＰ内の最後のシンボル（すなわち、長さｐのパターンに対する第ｐのシンボル）のすべての発生を示す第１の列Ｆ内の範囲の索引を判定することを伴う。ステップ２６１４は、第１および第２のプールからの識別子にアクセスして、範囲の第１および最後の索引ｈおよびｚにおけるランクを計算することを伴い、これらのランクは、パターンＰ内で範囲のシンボル（すなわち、Ｐ内の第（ｐ－ｉ）のシンボル、ここでループカウンタｉは最初に１に等しい）に先行するシンボルの値に対して判定される。ステップ２６１６で、各索引におけるランクを使用して、先行するシンボル（第（ｐ－ｉ）のシンボル）の値が生じる第１の列Ｆ内の新しい範囲の索引を計算する。決定ブロック２６１８で、ランク$r_{h-1}$および$r_z$を比較して、これらが等しいかどうかを検査

する。ランクが等しいと判定された場合（Ｙ）、方法２６００はステップ２６２０へ進み、ストリングＳ内にパターンＰの発生が存在しないことを示す０のカウントを出力する。ランクが等しくない場合（Ｎ）、方法２６００は決定ブロック２６２１へ進み、ｉがｐ－１（パターンの末端を示す）に等しいかどうかが検査される。ｉがｐ－１に等しい場合（Ｙ）、方法２６００はステップ２６２２へ進み、ここでパターンＰのカウントがｚ－ｈ＋１であると判定される。ｉがｐ－１に等しくない場合（Ｎ）、方法２６００はステップ２６２３へ進み、ここでループカウンタｉが１だけ増分され、プロセスはステップ２６１４へ戻り、ステップ２６１４～２６１８を少なくとももう一度繰り返す。ステップ２６１２～２６１６は、Ｐの第１のシンボルに到達し、カウントがステップ２６２２で出力されるまで、またはストリングＳ内にパターンＰの発生が存在しないと判定され、０のカウントがステップ２６２０で出力されるまで、繰り返される。

【０２４１】

ビットストリング内の各ビットは、ビット値およびビット位置を有する。各カウンタシンボルは、直列化されたカウンタアレイであるカウンタシンボルストリング内に記憶することができる。各カウンタシンボルは、特有のビット値、たとえば「１」を有するビットストリングＬ内のすべてのｗ個のビットに対するビットの数のランニングカウントを示すｂ個のカウンタビットのストリングによって表すことができる。ステップ２６１２は、パターンＰの第ｐのビット値の発生のＦ内の範囲を画定する第１の位置ｈおよび最後の位置ｚを判定することを伴うことができ、ここで範囲はｈおよびｚを含む。ステップ２６１４は、ストリングＬ内の位置ｈ－１で、パターンＰの第（ｐ－ｉ）のビット値のランク$r_{h-1}$を計算することを伴うことができ、ここでループカウンタｉは最初に１である。ステップ２６１４は、ストリングＬ内の位置ｚで、パターンＰの第（ｐ－ｉ）のビット値のランク$r_z$を計算することをさらに伴うことができる。$r_{h-1}$が$r_z$に等しい場合、メッセージ内のパターンの発生のカウントを０に設定することができる。ステップ２６１６は、ｈを第（ｐ－ｉ）のビット値のＦ内の第（$r_{h-1}+1$）のインスタンスの索引に設定すること、およびｚを第（ｐ－ｉ）のビット値のＦ内の第$r_z$のインスタンスの索引に設定することを伴うことができ、ここで新しいｈおよびｚは、新しい範囲を画定する。ｈおよびｚが再計算された後、ループカウンタｉを１だけ増分することができ、ステップ２６１４～２６１６が、ｉ＝ｐ－１になるまで繰り返される。ステップ２６１４～２６１６の繰返しは、連続して実行することができ、または繰返しは、たとえば上記で説明した照会木最適化方法を使用して、並列反応で実行することができる。

【０２４２】

各プールは、固体、液体、または固体の形態を有することができ、複数の識別子核酸分子を形成することによって形成することができる。各識別子は、それぞれのビット位置に対応することができ、Ｍ個の選択された成分核酸分子を物理的に組み立てることによって形成することができる。Ｍ個の選択された成分の各々は、Ｍ個の異なる層に分離された別個の成分核酸分子のセットから選択することができる。識別子は、ビットストリングを表すように、第１のプール内に収集することができ、したがってビット値は、第１のプール内の対応する識別子の有無によって示される。同様に、識別子の第２のプールは、各々カウンタストリング内のビットを表す識別子を収集することによって形成することができる。いくつかの実装形態では、第１のプールは第２のプールと同じであり、他の実装形態では、第１のプールおよび第２のプールは別個である。

【０２４３】

いくつかの実装形態では、第１のプール内の対応する識別子の物理的な存在は、ビット値１を示し、対応する識別子の物理的な不在は、ビット値０を示す。各カウンタシンボルは、ｂ個のカウンタビットのストリングによって表すことができ、ｂは、関数$\log_2(n+1)$の上限によって判定することができ、ここでｎはビットストリングの長さである。ｂ個のカウンタビットからなる各ストリングは、特有の値、たとえば１または０を有するビットストリング内のすべてのｗ個のビットに対するビットの数のランニングカウントを表すことができる。カウンタシンボルストリングは、ｎをｗで割った上限個のカウンタ

シンボルを含むことができ、ｂにｎをｗで割った上限を掛けた値に対応する長さを有するカウンタビットのストリングによって表すことができる。いくつかの実装形態では、初期カウンタシンボルは、値０を有し、これは、すべて値０を有するｂ個のカウンタビットのストリングによって表される。いくつかの実装形態では、カウンタストリングブロックサイズｗは、ビットストリングブロックサイズｂに等しく設定することができる。カウンタストリングブロックサイズｗは、１に設定することができる。いくつかの実装形態では、ステップ２６０２は方法２６００から省略される。

【０２４４】

いくつかの実装形態では、識別子の第１のプールは、Ｌ内のビットのブロックに変換された隣接して順序付けられた識別子のブロックに対応するコードワードを表すストリングＬの変換を表す。第１のプール内の識別子の有無は、ビットストリング内のいずれのビット値とも直接相関しない。コードワードは、固定数の可能な固有の識別子からの、固定数の固有の識別子に対応することができる。追加の情報を記憶することができる。たとえば、第１および第２のプールからの識別子の書込み、アクセス、および読取りのエラーの検出および補正で使用するために、追加の情報を記憶することができる。追加の情報は、第１または第２のプールの識別子に記憶することができる。

【０２４５】

単にビット値１または０などの長さ１のパターンの場合、ステップ２６１４～２６１８を省略することができる。「０１」または「１１」などの長さ２のパターンの場合、繰返しが必要とされないため、ステップ２６１８を省略することができる。方法は、第１の列Ｆ内の範囲が存在しないと判定された場合、方法を停止することをさらに伴うことができ、カウントは０として返される。範囲が存在しない場合、パターンＰのシンボルはストリング内に存在せず、したがってＰの発生は０である。

【０２４６】

いくつかの実装形態では、識別子の第３のプールが取得され、第３のプール内の識別子は、ＢＷＴから導出された接尾辞アレイＳＡを表す。ＳＡの各要素は、Ｌの対応する要素の索引を示す少なくともｌｏｇ$_2$（ｎ）個のビットのビットストリングによって表すことができる。方法２６００は、カウントが０より大きいと仮定すると、範囲の索引の最終値ｈおよびｚを含めて、ｈとｚとの間の接尾辞アレイの要素に対応する第３のプール内の識別子にアクセスすることによって、パターンの発生を位置付けることをさらに伴うことができる。これらの索引は、発生のカウントが返されるときの最後の範囲の索引とすることができる。これらのステップは、フェッチ演算として実行することができる。

【０２４７】

いくつかの実装形態では、方法２６００は、たとえばメッセージとすることができるビットストリングＳを表す識別子の第４のプールを取得することをさらに伴う。第１の場所および第１の場所を取り囲む隣接位置に対応する第４のプール内の識別子にアクセスすることによって、パターンＰの第１の場所のコンテキストを抽出することができる。

【０２４８】

図２６に関連して説明した計数方法に対する代替として、図２７は、例示的な実装形態によるストリングｓ内の２進パターンＰの発生を計数するカウント演算のための方法２７００について説明する流れ図を示す。パターンＰは、長さｐのビットを有し、ストリングｓは、長さｎのビットを有する。方法２７００はまず、２７０２で、パターンＰの最後のシンボルＰ［ｐ－１］が０に等しいかどうかを検査するように条件付きの決定ブロックを伴う。最後のシンボルが０に等しい場合（Ｙ）、ステップ２７０４で、第１の索引ｈが０（たとえば、Ｂｕｒｒｏｗｓ－Ｗｈｅｅｌｅｒ行列の第１の列内の第１の位置）に等しく設定され、最後の索引ｚは、＃０（ストリングｓ内の０の数）から１を引いた値に等しく設定される。この場合、第１および最後の索引ｈおよびｚは、各々０から始まる接尾辞の範囲の境界を定めることができる。Ｐの最後のシンボルが０に等しくない場合（Ｎ）、ステップ２７０５で、第１の索引が＃０に等しく設定され、最後の索引がｎ－１に等しく設定される。この場合、第１および最後の索引は、各々１から始まる接尾辞の範囲の境界を

定めることができる。

【０２４９】

方法２７００は、ステップ２７０６で、ループカウンタｉをｐ－２に等しく設定することをさらに伴う。このループカウンタは、発生が存在するまで（すなわち、第１の索引が最後の索引より小さいまたはそれに等しいとき）、方法２７００がＰを逆方向に走査することを可能にすることができる。２７０８で、決定ブロックが実施され、これは、第１の索引が最後の索引より小さいまたはそれに等しいこと、およびループカウンタｉが０より大きいまたはそれに等しいことに関して条件付きである。決定ブロック２７０８の条件が満たされなかった場合（Ｎ）、ブロック２７１０で、出力カウントはｚ－ｈ＋１に等しいと判定される。決定ブロック２７０８の条件が満たされた場合（Ｙ）、ステップ２７１１で、ストリングｓのＢｕｒｒｏｗｓ－Ｗｈｅｅｌｅｒ変換（$BWT_s$）上の第１の索引ｈに先行する索引（第１の索引から１を引いた値）で、シンボル値１に対して、第１のランク$r_{h-1}$が実行される。$BWT_s$上の最後の索引ｚに等しい位置で、シンボル値１に対して、最後のランク$r_z$が実行される。

【０２５０】

ステップ２７１２で、パターンＰの第ｉのシンボルが０に等しいことに関して条件付きで、別の決定ブロックが実行される。条件が満たされた場合（Ｙ）、ステップ２７１４で、第１の索引ｈは、現在の第１の索引から第１のランク$r_{h-1}$を引いた値より１小さい値に等しくなるように再計算され、最後の索引ｚは、現在の最後の索引から最後のランク$r_z$を引いた値より１小さい値に等しくなるように再計算される。ステップ２７１４で、新しい第１の索引ｈは、$BWT_s$上の現在の第１の索引に先行する索引で、シンボル値０に対して、現在の第１の索引とランクとの差として計算することができ、新しい最後の索引ｚは、$BWT_s$上の現在の最後の索引で、シンボル値０に対して、現在の最後の索引とランクとの差として計算することができる。決定ブロック２７１２の条件が満たされない場合（Ｎ）、これはパターンＰの第ｉの値が１であることを意味し、ステップ２７１５で、第１の索引ｈは、ｓ内の０の数と第１のランク$r_{h-1}$との和に等しくなるように再計算される。最後の索引ｚは、ｓ内の０の数と最後のランク$r_z$との和より１小さい値に等しくなるように再計算される。ステップ２７１６で、ループカウンタは１だけ減分され、方法は決定ブロック２７０８へ戻る。ブロック２７０８の条件がまた満たされる場合、ステップ２７１１～２７１６は繰り返される。

【０２５１】

方法２７００は、ステップ２７１１で、１ビットに対するランク演算、したがって表記$rank_1$を特別に実行する。これは、方法２６００がランク演算をどのように実行するかとはわずかに異なり、方法２６００では、パターンＰ内の先行するビット値に対してランクが判定される。しかし、ビット値１に対する位置ｘのランク$rank_1$（ｘ）は、以下の等式によって、ビット値０に対する位置ｘのランク$rank_0$（ｘ）に容易に変換することができるため、どちらの方法も同じ結果をもたらす。

$rank_0(x) = x - rank_1(x) + 1$　等式２

$rank_1(x) = x - rank_0(x) + 1$　等式３

したがって、ステップ２７１１は、別法として、$rank_0$を実施することができる。

【０２５２】

図２７の方法２７００のステップ２７１１は、２つのランク演算の実行を伴う。このステップについて、例示的な実装形態による２つのアクセス演算および２つの読取り演算を介して２つのランク演算を並列に実行する方法２７１１として、図２８でさらに詳細に説明する。図２８の方法２７１１は、方法２７００の決定ブロック２７０８に従って繰返し実行される。方法２７１１は、図１８および図１９に関連して説明した例示的な符号化方式からの特定のパラメータを使用するが、代替のパラメータを有する他の符号化方式をこれらの方法に適用することもできることを理解されたい。

【０２５３】

方法２７１１のステップ２８０２は、パラメータｘを現在の第１の索引（アルゴリズム

１から）より１小さい値に等しく設定することを伴う。ステップ２８０２は、$q_{first}^{BWT}$を、ｘを９６（各コンテナに符号化されたビットの数）で割った値の下限（それより小さい最大の整数）に等しく設定することをさらに伴い、ここで$q_{first}^{BWT}$は、$BWT_s$を記憶するライブラリ内のコンテナの索引であり、特有のコンテナは、$BWT_s$の第ｘのビットを記憶する。パラメータ$r_{first}^{BWT}$が、ｘおよび９６の係数（残余）に等しく設定され、ここでパラメータは、そのコンテナ内の第ｘのビットのオフセットを示す。カウンタアレイｃに対して、類似のパラメータが算出される。$q_{first}^{C}$は、$q_{first}^{BWT}$の下限を７（１つのコンテナ当たりのカウンタの数）で割った値に等しく設定され、ここで$q_{first}^{C}$は、ｃの第$q_{first}^{BWT}$のカウンタを含むｃを記憶するライブラリ内のコンテナである。$r_{first}^{C}$は、カウンタのそのコンテナ内のオフセットを示す$q_{first}^{BWT}$および７の係数に等しく設定される。

【０２５４】

方法２７１１のステップ２８０４は、パラメータｙを最後の索引（アルゴリズム１から）に等しく設定すること、ならびに$q_{last}^{BWT}$、$r_{last}^{BWT}$、$q_{last}^{C}$、および$r_{last}^{C}$によって示されるパラメータｙに対して、ステップ２８０２と同じＢＷＴおよびカウンタアレイのパラメータを設定することを伴う。ステップ２８０４は、$BWT_s$の第ｘのビットを含むコンテナおよびそのコンテナの前の１のカウンタを含むコンテナを識別する照会$Q_{first,BWT}$および$Q_{first,C}$を作成することを伴う。これらの照会はコンテナを識別し、したがって照会は、それぞれ塩基３の７桁における$q_{first}^{BWT}$および塩基３の７桁における$q_{first}^{C}$を表す成分を有する２つのパスとすることができる。ステップ２８０６は、それぞれ$q_{first}^{BWT}$および$q_{first}^{C}$に関して特殊化された照会$Q_{last,BWT}$および$Q_{last,C}$を同様に作成することを伴う。ステップ２８０８は、$BWT_s$を記憶する標的ライブラリに対して合同で$Q_{first,BWT}$および$Q_{last,BWT}$の第１のアクセス演算（論理ＯＲ）、ならびにカウンタアレイｃを記憶する標的ライブラリに対して合同で$Q_{first,C}$および$Q_{last,C}$の第２のアクセス演算を、並列に実行することを伴う。いくつかの実装形態では、合同の照会は各々７つの成分の４つのパスとすることができ、πは２に等しい（たとえば、ＰＣＲの場合）ため、これらのアクセス演算は、各々多くとも２つのコンテナ（これらが等しい場合は１つのコンテナとすることもできる）を取り出し、１６個のアクセサ（７の上限を読取りに対する技術的制限πで割った値に４を掛けた値）を必要とする。ステップ２８１０は、アクセス演算によって選択された識別子のセットを併合すること、およびこれらのセットを読み取ることを伴う。いくつかの実装形態では、これらの識別子は、４つ以下のコンテナ内に含まれ、各コンテナは、２８個以下の識別子を含み、したがってこれらは多くとも合計４×２８＝１１２である。

【０２５５】

ステップ２８１２は、パラメータ$n_{first}^{BWT}$で、$Q_{first,BWT}$に等しい長さ７成分の先導パスと、$r_{first}^{BWT}$より小さいまたはそれに等しい最後の成分とを有する識別子の数を計数することを伴う。同様に、パラメータ$n_{last}^{BWT}$は、$Q_{last,BWT}$に等しい長さ７成分の先導パスと、$r_{last}^{BWT}$より小さいまたはそれに等しい最後の成分とを有する識別子の数に等しく設定される。ステップ２８１４は、パラメータ$n_{first}^{C}$を、$q_{first}^{C}$に等しい長さ７成分の先導パスを有する識別子を記憶するコンテナ内の第$r_{first}^{C}$のカウンタに対応する値に等しく設定することを伴う。パラメータ$n_{last}^{C}$も同様に設定される。ステップ９で、出力、第１のランクＲＦが、$n_{first}^{BWT}$および$n_{first}^{C}$の和に等しく設定され、別の出力、最後のランクＲＬが、$n_{last}^{BWT}$および$n_{last}^{C}$の和に等しく設定される。

【０２５６】

第１および最後の索引によって境界が定められた範囲Ｒ内のストリングｓの接尾辞上に構築された接尾辞アレイｓａ［０，ｎ－１］内に存在する発生の取出しに関して、フェッチ演算は、接尾辞アレイを記憶するライブラリに対して、それらの発生を収容するコンテナと同程度の長さ７成分のパスを指定することによって実施することができる。符号化方

式の例では、すべてのコンテナが、７つの接尾辞位置を含み、したがって２＋（最後の索引－第１の索引＋１）／７以下のパスによって、フェッチ演算を実行することができる。これらのパスは、コンテナの連続する範囲を識別することができるため、成分を共用することができる。必要とされるより多くの識別子を取り出すスーパークエリを実施して、後処理算出段階への識別子のフィルタリングを遅らせることができる。最小整数ｚは、（最後の索引－第１の索引＋１）より大きい３$^{z}$×７に等しいサイズＳ（ｚ）を画定することができ、ここでｚは０、１、．．．、７に等しい。範囲Ｒは、サイズＳ（ｚ）の多くとも２つの範囲によって含むことができ、これらは、（７－ｚ）個の成分からなる２つの部分パス照会によって画定することができる。スーパークエリは、（７－ｚ）／πの上限に等しい数のアクセサによって、場合により１つの複製演算によって実行することができ、パターン発生の上位セットを得ることができる（７より多い発生の場合、乗法因子３による）。次いで、２つの照会に由来する識別子の併合に対して、読取り演算を実行することができる。

【０２５７】

ここまで、カウント演算について、２進ストリングを使用する例で説明してきたが、カウント演算は、任意のストリングを記憶するＤＮＡで実施することもできる。任意のアルファベットから引き出されるストリングｓが、図１６および図１７に示すＢｕｒｒｏｗｓ－Ｗｈｅｅｌｅｒ変換および接尾辞アレイに対する基本として働いた。ストリングＢＷＴ（ｓ）および追加のデータ構造を活用することによって、本開示は、任意のアルファベットから引き出されたパターンの探索を提供する。図２９は、例示的な実装形態による任意のシンボルストリング上でカウント演算を実行する方法２９００のための流れ図を示す。ステップ２９０２は、識別子の第１のプールを取得することを伴い、識別子は、シンボルストリング（ｓ）のＢｕｒｒｏｗｓ－Ｗｈｅｅｌｅｒ変換Ｌ（ＢＷＴ行列の最後の列）を表し、ここでシンボルストリング内の各シンボルは、シンボル値のセットから得られる。ステップ２９０４は、識別子の第２のプールを取得することを伴い、各第２のプールは、ＢＷＴ（ｓ）Ｌから導出されたカウンタシンボルストリングを表し、特定のシンボル値を有するＬ内のシンボルの数のランニングカウントを表す。ステップ２９０６は、第１のプールおよび第２のプールセットからの識別子に選択的にアクセスすることによって、シンボルストリング内の特定のシンボルパターンのカウントを取得することを伴う。方法２９００に続いて、２進ストリングに関して図２６～図２８に説明した方法を、ビットストリングおよび対応するカウンタストリングを表すプールの各対に適用することができる。

【０２５８】

たとえば、ステップ２９０６は、ｓに対するＢｕｒｒｏｗｓ－Ｗｈｅｅｌｅｒ変換行列の第１の列Ｆを再構築することを伴うことができる。一連のプローブを使用して、Ｌ内の対応する各シンボル値の発生の総数を表す第２のプールの各々の最後のカウンタシンボルから、識別子にアクセスすることができる。対応する各シンボル値の発生のこの総数を使用して、Ｆを再構築することができる。Ｆが分かると、２９０６は、長さｐを有するパターン内の最後（第ｐ）のシンボル値を有するＦ内の位置の範囲を判定することをさらに伴うことができる。Ｆ内の位置の前記範囲は、第１の位置ｈおよび最後の位置ｚによって、ｈおよびｚを含めて画定される。

【０２５９】

ステップ２９０６は、第ｐのシンボル後のパターン内の先行する各シンボルに対して、一連のプローブを使用して、範囲にすぐ先行する位置におけるＬ内の対応するシンボル値の第１のランク、および範囲の末端の位置におけるＬ内の対応するシンボル値の第２のランクを判定して、第１のプールおよび対応する第２のプールからの識別子核酸分子にアクセスすることと、第１のランクおよび第２のランクを使用して、パターン内の後続のシンボルに先行する対応するシンボルのインスタンスを有するＦ内の位置の範囲に、範囲を更新することとをさらに含むことができる。第１のランク$r_{h-1}$は、Ｌ内の位置ｈ－１におけるパターン内のそれぞれの先行するシンボル値であり、第２のランク$r_{z}$は、Ｌ内の位置ｚにおけるパターン内のそれぞれの先行するシンボル値である。範囲を更新することは

、Ｆ内のパターン内のそれぞれの先行するシンボル値の第（$r_{h-1}+1$）のインスタンスの位置にｈを設定すること、およびＦ内のパターン内のそれぞれの先行するシンボル値の第$r_z$のインスタンスの索引にｚを設定することを含む。パターンの発生のカウントは、第１および第２のランクまたは第１および最後の索引の最終値に基づいて設定することができる。たとえば、カウントは、第１および第２のランクの最終値間の差である。第ｐのシンボルまたはあらゆる先行するシンボルに対して、第１および第２のランクが互いに等しい場合、カウントは０に設定される。

【０２６０】

シンボルストリングＬは、Ｌの長さに等しい長さのｒ個のビットストリングに変換することができ、ここでｒは、シンボル値のセット内のシンボル値の数を表す。１つのビット値（たとえば、１）でＬ内の$R_v$のすべての発生を表し、他のビット値（たとえば、０）ですべての他のシンボル値の発生を表すことによって、ｖ＝１、２、．．．、ｒの場合、各シンボル値$R_v$に対して、ビットストリング$L_v$を作成することができる。たとえば、シンボルストリングＳ＝「ＢＡＢＢＯ」を考慮すると、ビットストリング$S_A$＝０１０００、$S_B$＝１０１１０、および$S_O$＝００００１を作成することができる。すべてのビットストリングの中の１の総数は、元のシンボルストリングの長さに等しい。いくつかの実装形態では、ｒ個の第１のプールが存在し、各々ビットストリング$L_v$に対応する。$L_v$に対応する第１のプールを使用して、パターン内のシンボル値$R_v$の第１および第２のランクを判定することができる。

【０２６１】

カウンタシンボルは、各ビットストリングから導出することができ、したがって各カウンタシンボルは、特有のビット値（たとえば、１）を有する対応するビットストリング内のすべてのｗ個のビットに対するビットの数のランニングカウントを示すｂ個のカウンタビットのストリングによって表される。いくつかの実装形態では、ｗの値は、ｂの値に等しく設定される。他の実装形態では、ｗは１の値に設定される。

【０２６２】

各プールは、固体、液体、または固体の形態を有することができ、複数の識別子核酸分子を形成することによって形成することができる。各識別子は、それぞれのビット位置に対応することができ、Ｍ個の選択された成分核酸分子を物理的に組み立てることによって形成することができる。Ｍ個の選択された成分の各々は、Ｍ個の異なる層に分離された別個の成分核酸分子のセットから選択することができる。識別子は、ビットストリングを表すように、第１のプール内に収集することができ、したがってビット値は、第１のプール内の対応する識別子の有無によって示される。同様に、識別子の第２のプールは、各々カウンタストリング内のビットを表す識別子を収集することによって形成することができる。いくつかの実装形態では、第１のプールは第２のプールと同じであり、他の実装形態では、第１のプールおよび第２のプールは別個である。

【０２６３】

識別子のプールは、識別子の有無が$L_v$内のいずれのビット値とも直接相関しないが、ｖ＝１、２、．．．、ｒの場合、コードワードと呼ばれる隣接して順序付けられた識別子のブロックを、$L_v$内のビットのブロックに変換させることができるように構成することができる。コードワードは、固定数の可能な固有の識別子からの、固定数の固有の識別子に対応することができる。追加の情報を記憶することができる。たとえば、第１および第２のプールからの識別子の書込み、アクセス、および読取りのエラーの検出および補正で使用するために、追加の情報を記憶することができる。追加の情報は、第１または第２のプールの識別子に記憶することができる。

【０２６４】

いくつかの実装形態では、方法２９００は、ＢＷＴから導出された接尾辞アレイＳＡを表す識別子のプールを取得することをさらに伴う。ＳＡの各要素は、Ｌの対応する要素の索引を示す少なくともｌｏｇ$_2$（ｎ）個のビットのビットストリングによって表すことができる。方法２９００は、カウントが０より大きいと仮定すると、接尾辞アレイを表すプ

ール内の識別子にアクセスすることによって、パターンの発生を位置付けることをさらに伴うことができる。これらのステップは、フェッチ演算として実行することができる。いくつかの実装形態では、方法２９００は、たとえばメッセージとすることができる任意のストリングを表す識別子のプールを取得することを伴う。第１の場所および第１の場所を取り囲む隣接位置に対応する第４のプール内の識別子にアクセスすることによって、パターンＰの第１の場所のコンテキストを抽出することができる。

【０２６５】

サイズ｜Σ｜の任意のアルファベットΣから引き出された長さｎのシンボルのストリングｓを考慮する。図１６および図１７に示す「ａｂｒａｃａｄａｂｒａ＄」というストリングは、そのような任意のストリングの一例である。図２９に関連して説明した前述の方法では、ストリングｓ内の長さｐのパターンＰの探索は、ストリング接頭辞ｓ［０，ｘ］（第１の索引から索引ｘまでのストリングｓのサブセット）内のシンボルσの発生を計数する概略的なｒａｎｋσ（ｘ）演算の実装を介して、ｓのＢＷＴの接頭辞内のシンボルを計数することに依拠することを論じた。この方法は、探索されるパターン内のシンボルと同数のｒａｎｋσ演算を実行する必要がある。なぜなら、ビット値「１」を有する２進ストリング$S_\sigma$の特性に対して１つのｒａｎｋ１（ｘ）演算を実行することによって、ｒａｎｋσ（ｘ）を実施することができるからであり、ここでｓはシンボルσを有する。

【０２６６】

以下の実験例は、より少ないｒａｎｋσ演算を実行すること、および低い重みのコードブックで機能する手法を利用することを目的とする。この実装形態は、パターンＰの探索におけるシンボルのグループ化を活用し、アルファベットを事実上拡大し、繰返しの数を低減させる。

【０２６７】

この例では、ストリング$S^k$［０：ｎ－１］は次のように画定される。ｋは、１から始まり、ストリング長さｎによって上限が制限される整数パラメータであり、マクロシンボル$S^k$［ｉ］は、ＢＷＴを含む行列内の第ｉの行の最後のｋ個のシンボルを占有するサブストリングである。各ストリング$S^k$は、ｎ×ｋバイトに記憶することができる。この例では、「ａｂｒａｃａｄａｂｒａ＄」という入力ストリングを使用して、以下のサブストリングが得られる。$S^1$は、ＢＷＴ行列内の行の最後のｋ＝１個のシンボルがちょうど最後の列Ｌ、すなわちＢＷＴ（ｓ）であるため、ｓのＢＷＴ、すなわちＢＷＴ（ｓ）である。$S^2$は、最後のｋ＝２個のシンボルが各行から得られるため、「ｒａ　ｂｒ　ａｄ　ａ＄　ｂｒ　ａｃ　ｄａ　＄ａ　ｒａ　ｃａ　ａｂ　ａｂ」に等しい。読みやすくするために空間が含まれており、２個のシンボルの各グループは、固有の「マクロシンボル」であると考えられる。したがって、マクロシンボル$S^2$［３］は「ａ＄」であり、$S^2$は、各々２つの文字を有するｎ＝１２個のマクロシンボルからなり、合計２４バイトである。$S^3$は、最後のｋ＝３個のシンボルが各行から得られるため、「ｂｒａ　ａｂｒ　ｃａｄ　ｒａ＄　ａｂｒ　ｒａｃ　ａｄａ　ａ＄ａ　ｂｒａ　ａｃａ　ｄａｂ　＄ａｂ」に等しい。$S^3$は、各々３つの文字を有する１２個のマクロシンボルからなり、合計３６バイトである。これらのストリングは、各ｋに対して最大ｎまで繰返し構築することができる。

【０２６８】

$S^k$個のストリングは、必ずしも明示的に構築されるとは限らず、次のように導出することもできる。新しいストリング

【化２】

$$\hat{S}^k$$

は、

【化３】

$$\hat{S}^k[i]$$

がＳ$^k$［ｉ］［１］に等しくなるように画定することができ、これは、Ｓ$^k$のすべてのマクロシンボルに対して、マクロシンボルの第１のシンボル、したがってＢＷＴ行列内の最後の列Ｌから距離ｋをあけたシンボルのみが維持されることを意味する。Ｓ$^k$［ｉ］の第ｉのマクロシンボルは、すべてのストリングの第ｉの（単一の）シンボル

【化４】

$$\hat{S}^k[i]\hat{S}^{k-1}[i]\hat{S}^{k-2}[i]\ldots\hat{S}^1[i]$$

を連結することによって得られる。低減された空間占有率は、各Ｓ$^k$によって記憶されるｋ×ｎ個のシンボルより少ないｎ個のシンボルのみを記憶する

【化５】

$$\hat{S}^k$$

によって与えられる。

【０２６９】

パターン探索の場合、Ｓ$^k$個のストリングにわたってｒａｎｋ$_\alpha$演算を実行しなければならず、ここでαは、長さｋのシンボルのマクロシンボルである。符号化方策は、

【化６】

$$\hat{S}^k$$

個のストリングの単一のシンボルに対して実施することができ、この方策は、シンボルストリング「ＢＡＢＢＯ」に対する上記の例を模倣する。具体的には、

【化７】

$$\hat{S}^k$$

個のストリングが、各々ｂ個のコンテナのブロックに区画化される（実験例では、各コンテナが、１２バイト、したがってｓの１２シンボルを符号化する）。カウンタｃ$_\alpha{}^k$は、

【化８】

$$\hat{S}^k$$

内のその第ｋのブロックまで、すべてのマクロシンボルαの発生の数に対して維持される。Ｓ内で生じるマクロシンボルα’＜αの総数を、カウンタ$c_\alpha{}^k$に加えることができる。この特徴は、発生の数とともに追加のカウンタアレイＣを記憶する必要がないことを可能にすることができる。１つのブロック当たり、多くとも｜Σ｜$^k$個のカウンタが存在することができる。

【０２７０】

いくつかの実装形態では、長さｋのすべてのサブストリングαがｓ内で生じるとは限ら

ず、したがってｃ$_{\alpha}{}^{k}$に記憶する必要はない。存在するまたは存在しないサブストリングを追跡して、本明細書に記載する符号化方式を介してＤＮＡに、または本明細書に記載するＤＮＡに基づくデータ記憶とコンピュータの内部メモリとの両方を導入する複合記憶方式によって、対応するデータ構造のみを記憶することができる。複合記憶方式は、ストリングｓ内の既存の（マクロ）シンボルと範囲［０，．．．］内の整数との間の再マッピングを実施することができる。本当のアルファベット

【化９】

$$\hat{\Sigma}_k$$

は、ｓ内に現れるシンボルを示すことができる（連続する整数によって適切に符号化される）。任意のｋに対してｎまたは｜Σ｜$^{k}$の最小値より少ないまたはそれに等しい本当のアルファベットの

【化１０】

$$\hat{\Sigma}_k$$

個のシンボルが存在することができる。この再マッピングは、ランクデータ構造に基づいて内部メモリまたは圧縮解内に維持されたハッシュ表（たとえば、カッコウハッシング）によって実施することができる。

【０２７１】

例として、パターンＰがストリング「ａｂｒａｃａｄａｂｒａ」内の「ｄａｂ」に等しいものとする。長さ１および２のマクロシンボルに対して、前述のデータ構造が導入され、Ｐに対する探索が、Ｐ上で逆方向に演算する２つの段階に分割される。まず「ａｂ」の探索、第２に「ｄ」の探索が行われる。最初に０に等しい第１の索引、および最後に１１に等しい最後の索引から始まり、接頭辞ａｂを有する行が判定される。この第１のステップは、ストリングＳ$^{2}$にアクセスすること、ならびにこのストリングにわたって、０に等しい演算ｒａｎｋ$_{ab}$（ｆｉｒｓｔ－１）および２に等しいｒａｎｋ$_{ab}$（ｌａｓｔ）を実行することを伴う。これらのランク演算は、ストリングＳ$^{2}$およびそのカウンタアレイｃ$_{ab}{}^{2}$にわたって、２つの並列のアクセス演算および２つの並列の読取り演算によって実行することができる。「ａｂ」より小さいストリングの数は、ｓ内の２であり、したがって索引に対する新しい値は、ｆｉｒｓｔ＝２＋ｒａｎｋ$_{ab}$（ｆｉｒｓｔ－１）＝２およびｌａｓｔ＝２＋ｒａｎｋ$_{ab}$（ｌａｓｔ）－１＝３である。図１６に見ることができるように、接頭辞ａｂを有する行は２つであり、ＢＷＴ行列の行２および３で生じる。第２のステップは、接頭辞「ｄａｂ」、したがって「プリペンド」するシンボル「ｄ」から「ａｂ」を有する行を判定することを伴う。このステップは、Ｓ$^{1}$にアクセスし、このストリングにわたって、演算ｒａｎｋ$_{d}$（ｆｉｒｓｔ－１）＝０およびｒａｎｋ$_{d}$（ｌａｓｔ）＝１を実行することによって実行される。「ｄ」より小さいストリングの数は、ｓ内の９であり、したがって索引に対する新しい値は、ｆｉｒｓｔ＝９＋ｒａｎｋ$_{d}$（ｆｉｒｓｔ－１）＝９およびｌａｓｔ＝９＋ｒａｎｋ$_{d}$（ｌａｓｔ）－１＝９である。新しい索引は同等であり、したがってプロセスを停止して、Ｐの発生のカウントを１に戻すことができる。この例では、一度に１つのシンボルで探索が実行された場合に必要とされるはずの３回ではなく、２回の繰返し／ステップを実行して、Ｐを探索する。類似の探索演算について、概略的なケースに関して図３２に関連して説明する。

【０２７２】

ストリング内のパターンの発生を計数するために、ＢＷＴ、カウンタアレイ、接尾辞アレイ、ならびにアクセスおよび読取り方法が実施されている。いくつかの実装形態では、ストリング自体は、ＤＮＡに記憶することができ、特定のパターンの場所でアクセスする

ことができる。探索方法を使用して、ストリング内のパターンの場所を発見する場合、ストリングを表す識別子プール内で、それらの場所および取り囲む近傍にアクセスすることができ、後に読み取ることができる。このプロセスは、たとえば特定のストリングパターンが生じるコンテキストを集めるために有用となりうる。

【０２７３】

識別子ライブラリを別個のコンテナ内へ複製および等分することができ、したがって複数のパターン探索を同時に並列で実行することができる。たとえば、１０、１００、１０００、またはそれよりも多くのパターン探索を並列に実行することができる。ＤＮＡシーケンサおよび固有のバーコードを使用して、異なる探索に属する識別子にラベル付けすることによって、異なる探索のランクおよびフェッチ演算を、読取り箇所で多重化することができる。

【０２７４】

上記では、ＢＷＴが順方向変換および逆方向変換を含むことができることについて論じた。上記の例によって、ＢＷＴに対する分類された循環シフト行列Ｍ’の各行が、シンボル＄が添えられた入力ストリングＳ（すなわち、ストリングＳ＄）の置換を収容することが示されている。同様に、Ｍ’の各列は、Ｓ＄の置換を収容する。特に、例示的なストリング「ａｂｒａｃａｄａｂｒａ」に対する第１の列Ｆは、アルファベット順に分類された「ａａａａａｂｂｃｄｒ」であり、元の入力ストリングの最善に圧縮可能な変換を表す。Ｆは転置可能でなく、したがって最後の列Ｌは、可逆性および圧縮性のために、ＢＷＴとして使用される。

【０２７５】

Ｍ’のｉのすべての行は、左向きの循環シフト、Ｓの接尾辞Ｓ［$k_i$，ｎ－１］、特殊シンボル＄、および接頭辞Ｓ［１，$k_i$－１］のため、３つの部分に分解することができる。たとえば、図１６で、Ｍ’の行２は接頭辞「ａｂｒａ」を有し、それに＄が続き、接尾辞「ａｂｒａｃａｄ」を有する。ＢＷＴ行列の第１の特性として、行ｉがＳ＄を表すことを除いて、最後の列Ｌ［ｉ］のシンボルは、ストリングＳ内で第１の列Ｆ［ｉ］のシンボルに先行し、したがってＦ［ｉ］＝Ｓ［１］およびＬ［ｉ］＝＄である。この特性は、ＭおよびＭ’のすべての行がＳ＄の左循環シフトであるという性質の結果であり、したがって各行の第１および最後のシンボルを得ることによって、ストリングＳにわたって、最後のシンボル（Ｌ内）の直後に第１のシンボル（Ｆ内）が位置する。ＢＷＴ行列の第２の特性は、Ｌ内の同じシンボルｃのすべての発生が、Ｆ内と同じ相対順序を維持することである。この特性は、Ｌ内のシンボルｃの第ｋの発生がＦ内のシンボルｃの第ｋの発生に対応することを意味する。「ＬＦマッピング」と呼ばれる方法は、これらの特性を活用して、そのＢＷＴ（ｓ）＝（Ｌ’，ｒ）からＳを再構築することができる。ＬＦマッピングは、サイズｎのアレイとして表すことができる。シンボルＬ［ｉ］がシンボルＦ［ｊ］にマッピングされる場合かつその場合に限り、アレイの第ｉのエントリＬＦ［ｉ］はｊに等しい。Ｌ［ｉ］がＬ内のシンボルｃの第ｋの発生である場合、Ｆ［ＬＦ［ｉ］］はＦ内のｃの第ｋの発生である。

【０２７６】

図３０は、例示的な実装形態による列ＬからＬＦマッピングを構築する方法３０００を示す。ステップ３００２で、方法３０００は、サイズ｜Σ｜＋１の補助ベクトル$V_A$を画定する。$V_A$は、シンボルまたは整数によって索引付けることができる。ステップ３００４で、ループの第１に、Ｌ内の各シンボルｃに対して、Ｌ内のその発生の数$n_c$を判定し、ステップ３００６で、$V_A$［ｃ］＝$n_c$を記憶する。次いで、ステップ３０１０で、ループの第２に、これらのシンボルごとの発生を累積和にし、したがってステップ３０１２で、新しい$V_A$［ｃ］が、ｃより小さいシンボルのＬ内の発生の総数を記憶し、すなわち$V_A$［ｃ］＝$\Sigma_{x<c} n_x$になる。このステップは、ステップ３００８で、２つの補助変数を採用することによって行われ、したがって全体的な作業空間は、約ｎのサイズのままである。$V_A$［ｃ］は、シンボルｃが生じるＦ内の第１の位置を与える。したがって、ループの最後が開始する前、$V_A$［ｃ］は、Ｌ内の第１のｃのＦ内の着地位置である（すべて

のアルファベットシンボルの第１の発生に対するＬＦマッピングが既知である）。最後に、ステップ３０１４で、ループの最後に、列Ｌを走査し、シンボルＬ［ｉ］＝ｃに遭遇したときはいつでも、ステップ３０１６で、ＬＦ［ｉ］＝$V_A$［ｃ］を設定する。この条件は、第１の時間に対してｃが満たされるときに適正であり、次いで$V_A$［ｃ］は、Ｌ内のｃの次の発生がＦ内の次の位置にマッピングされるように増分される（そのシンボルから始まるすべての行のＦ内の近接性を考慮する）。したがってアルゴリズムは、ＬＦ［ｉ］＝$\Sigma_{x<c} n_x + k$の不変量を維持し、その後、Ｌ内のｃのｋ回の発生が処理される。ステップ３０１８で、最後のＬＦ（ＬＦマッピングアレイ）の出力が生成される。

【０２７７】

上述したＬＦマッピングおよび基礎的な特性を考慮すると、Ｓは、変換された出力ＢＷＴ（ｓ）＝（Ｌ’，ｒ）から始まって、逆方向に再構築することができる。例示的な実装形態によるストリングＳの再構築のための方法３１００が、図３１に示されている。ステップ３１０２で、Ｌは、Ｌ’の位置ｒに＄を挿入することによって、ＢＷＴ（ｓ）から再構築される。ステップ３１０４で、ＬのＬＦマッピングが判定される。ステップ３１０４は、図３０の方法３０００を介して実行することができる。ステップ３１０６で、補助変数ｋおよびｉ（ループカウンタ）が設定される。変数ｋは、最初に０に設定され、ループカウンタｉは、最初にｎ－１に設定され、長さの最後の位置はＬ、Ｆ、およびｓである。決定ブロック３１０８で、方法３１００は、ループカウンタｉが０より大きいまたはそれに等しいかどうかを検査する。条件が満たされた場合（Ｙ）、繰返しはステップ３１１０を介して続行する。ステップ３１１０は、Ｍ’の第１の行が＄Ｓであると仮定すると、Ｓの最後のシンボル、すなわちＳ［ｎ－１］を選ぶことを伴い、これをＬ［０］で識別することができる。次いで、方法は、Ｓ内で１つのシンボルを一度に左へ動かし、上記の２つの特性を導入することによって続行し、第２の特性は、Ｌ（最初はＬ［０］）内で生じる現在のシンボルをＦ内の対応するコピーにマッピングすることを可能にし、次いで、第１の特性は、同じ行の末端（すなわち、Ｌ内のもの）でシンボルを得ることによって、Ｆ内のそのコピーに先行するシンボルを発見することを可能にする。この２重のステップは、Ｌで戻り、ｓにおける１つのシンボルの左向きの動きを可能にする。このプロセスをＳの始めまで繰り返すことで、このストリングを再構築することができる。決定ブロック３１０８の条件が満たされなくなった場合（Ｎ）、方法３１００はステップ３１１１で終了し、ここで元のストリングｓが完全に構築される。

【０２７８】

一例として、図１６を参照されたい。図１６では、Ｌ［０］＝Ｓ［ｎ－１］＝「ａ」になり、方法３１００のループが実行される（ステップ３１０８および３１１０を繰り返す）。ＬＦ［０］は、「ａ」から始まる第１の行、すなわち行１を指す。したがって、「ａ」のコピーがＦ［１］にＬＦマッピングされ（実際には、Ｆ［１］＝「ａ」）、Ｓ内の先行するシンボルはＬ［１］＝「ｒ」である。これら２つの基本ステップが、ストリングＳ全体が再構築されるまで繰り返される。前述の実行中の例を引き続き参照すると、Ｌ［１］＝「ｒ」が位置ＬＦ［１］＝１０のＦ内のシンボルにＬＦマッピングされる（実際には、Ｆ［１０］＝ｒ）。Ｌ［１］およびＦ［１０］は、それぞれ両方の列ＬおよびＦにおけるシンボル「ｒ」の第１の発生である。次いで、アルゴリズムは、Ｓ内の「ｒ」の先行するシンボルとして、シンボルＬ［１０］＝「ｂ」を得る。次いで、これらのステップが、ストリングＳの残余の先行する各シンボルに対して繰り返される。

【０２７９】

回転行列Ｍの行とストリングＳの接尾辞との間の全単射の対応関係、ならびにＳ上に構築される最後の列Ｌと接尾辞アレイとの間の関係について記載したように、これらの関係は、上記で論じたＦＭ索引の設計の中心である。ＦＭ索引は、効率的なサブストリング探索およびストリングサイズ約ｎのサイズの空間占有率を可能にする。したがって、３つの基本探索演算が、そのような索引付け探索の設計の根底にあり、ｃｏｕｎｔ（Ｐ）は、ストリングＰの接頭辞を有するＭ内の行の範囲［ｆｉｒｓｔ，ｌａｓｔ］を返し、ここで値（ｌａｓｔ－ｆｉｒｓｔ＋１）は、パターン発生の数を補償し、ｌｏｃａｔｅ（Ｐ）は、

Ｐが生じるストリングＳ内のすべての位置のリストを返し、ここでリストは、分類済みまたは未分類であり、ｅｘｔｒａｃｔ（ｉ，ｊ）は、ＦＭ索引内のその圧縮表現にアクセスすることによって、サブストリングＳ［ｉ，ｊ］を返す。たとえば、図１７で、パターンＰ＝「ａｂ」の場合、索引ｆｉｒｓｔ＝２およびｌａｓｔ＝３が、合計２つのパターン発生を返す。これら２つの行は、接尾辞Ｓ［０，１］およびＳ［７，：］に対応し、パターンＰの接頭辞を有する。

【０２８０】

図２６～図２９に関連して上述したように、行ｆｉｒｓｔおよびｌａｓｔの取出しは、２進探索を介して実施されるのではなく、列Ｌ、アレイＣ（Ｃ［ｃ］内でｃより小さいすべてのシンボルのｓ内の発生の数を計数する）、およびストリング接頭辞Ｌ［０，ｋ－１］内のシンボルｃの発生の数を報告する計数ｒａｎｋ（ｃ，ｋ）に効率的に対応する追加のデータ構造を導入する探索方法を使用する。すべてのデータ構造Ｌ、Ｃ、およびｒａｎｋを、圧縮した状態で記憶することができ、それでもなお、それらのエントリ、すなわちアクセスＬ［ｉ］もしくはＣ［ｃ］を効率的に取り出し、またはｒａｎｋ（ｃ，ｋ）で応答することができる。

【０２８１】

図３２は、例示的な実装形態による上記のアンサンブルを使用してｃｏｕｎｔ（Ｐ）を実施するように構成された方法３２００を示す。ステップ３２０２は、ループカウンタｉを、パターンＰ内の最後の位置を示すｐ－１に等しくなるように設定することを伴う。可変シンボルｃが、Ｐ［ｐ－１］によって示すように、Ｐ内の最後のシンボルに等しく設定される。ステップ３２０４で、第１の索引ｈは、Ｃ［ｃ］によって示すように、可変シンボルｃに対するカウンタエントリに等しく設定され、ここでＣは、シンボルストリングのＢｕｒｒｏｗｓ－Ｗｈｅｅｌｅｒ変換から導出されるカウンタアレイである。最後の索引ｚは、位置ｃ＋１のカウンタエントリより１小さい値に等しく設定される。第１および最後の索引ｈおよびｚは、ＢＷＴから導出される行列Ｍ’の第１の列Ｆ内の値の範囲の境界を定める。決定ブロック３２０６は、第１の索引ｈが最後の索引ｚより小さいかどうか、およびループカウンタｉが１より大きいまたはそれに等しいかどうかを検査する。両方の条件が満たされた場合（Ｙ）、ステップ３２０８で、可変シンボルｃが、パターンＰの先行するシンボルに設定される。新しい第１の索引ｈが、ｃに対するカウンタエントリと、シンボル値ｃに対する位置ｈ－１で算出されるＢＷＴのランクとの和に等しく設定される。新しい最後の索引ｚが、ｃに対するカウンタエントリと、シンボル値ｃに対する位置ｚで算出されたＢＷＴのランクとの和より１小さい値に等しく設定される。ループカウンタｉは、１だけ減分され、方法はステップ３２０４へ戻る。

【０２８２】

方法３２００の各繰返しステップ３２０４～３２０８は、次の不変量を保持する。第ｉの段階で、パラメータ「ｈ」は、接頭辞Ｐ［ｉ，ｐ－１］を有する分類された回転行列Ｍ’の第１の行を指し、パラメータ「ｚ」は、接頭辞Ｐ［ｉ，ｐ－１］を有するＭ’の最後の行を指す。最初に、不変量は構成によって真であり、第１の列Ｆに対して、Ｆ［Ｃ［ｃ］］は、ｃから始まるＭ’の第１の行であり、Ｆ［Ｃ［ｃ＋１］－１］は、ｃから始まるＭ’の最後の行である。図１７に関するＰ＝「ａｂ」の例を再び参照すると、初期条件は、Ｃ［ｂ］＝６およびＣ［ｂ＋１］＝Ｃ［ｃ］＝８であり、［６，７］は、接頭辞ｂを有する行の範囲であり、その前に、逆方向探索が開始する。

【０２８３】

後続の各繰返しにおいて、方法３２００は、接頭辞Ｐ［ｉ，ｐ－１］を有する行の範囲［ｈ，ｚ］を発見する。次いで、この方法は、次のように進むことによって、接頭辞Ｐ［ｉ－１，ｐ－１］＝Ｐ［ｉ－１］Ｐ［ｉ，ｐ－１］を有する行の新しい範囲［ｈ，ｚ］を判定することを伴う。第１に、適切に照会された関数ｒａｎｋを導入することによって、サブストリングＬ［ｈ，ｚ］内のシンボルｃ＝Ｐ［ｉ－１］の第１および最後の発生を判定する。具体的には、ｒａｎｋ（ｃ，ｈ－１）が、ｃのどれだけの発生がＬ内の位置ｈに先行するかを計数し、ｒａｎｋ（ｃ，ｚ）が、ｃのどれだけの発生がＬ内の位置ｚに先行

するかを計数する。次いで、これらの値を使用して、ｃのそれらの第１／最後の発生のＬＦマッピングを算出する。同等ＬＦ［ｉ］＝Ｃ［Ｌ［ｉ］］＋ｒａｎｋ（Ｌ［ｉ］，ｉ）が存在する。この同等は、ｒａｎｋ（ｃ，ｋ）を実施するデータ構造が密に記憶されるという条件で、ＬＦマッピングの算出を効率的かつ簡潔に行うことができることを意味する。図１７を再び参照して、前述のように、パターンＰ＝「ａｂ」および接頭辞Ｐ［２］＝「ｂ」を有するＭ’内の行の範囲［６，７］を考慮する。次に、前述のパターンシンボルＰ［１］＝「ａ」を選ぶと、Ｌ［０，ｈ－１］が「ａ」の１つの発生を収容し、Ｌ［０，ｚ］が「ａ」の３つの発生を収容するため、アルゴリズム５は、ｒａｎｋ（「ａ」，５）＝１およびｒａｎｋ（「ａ」，７）＝３を算出する。したがって、アルゴリズムは、ｈ＝Ｃ［「ａ」］＋ｒａｎｋ（「ａ」，５）＝１＋１＝２、ｚ＝Ｃ［「ａ」］＋ｒａｎｋ（「ａ」，７）－１＝１＋３－１＝３として、新しい範囲を算出し、これは、接頭辞パターンＰ＝「ａｂ」を有する行の隣接範囲である。最後の段階（すなわち、ｉ＝０）後、ｆｉｒｓｔおよびｌａｓｔは、接頭辞Ｐを有するすべての接尾辞を収容するＭ’の行の境界を定める。ｚ＜ｈの場合、パターンＰはＳで生じない。

【０２８４】

以下、手順ｌｏｃａｔｅ（Ｐ）を介したパターン発生の位置づけの実装形態について説明する。固定のパラメータμの場合、本発明者らは、ｊ＝０、１、２、．．．の場合、形式ｐｏｓ（ｉ）＝ｊμの位置で開始する接尾辞に対応するＭ’の行ｉをサンプリングする。そのような各対＜ｉ，ｐｏｓ（ｉ）＞は、一定時間内（行成分上）のメンバーシップ照会に対応するデータ構造Ｌに明示的に記憶される。次に、行索引ｒを考慮すると、Ｌ内のｒがサンプリングされた行である場合、値ｐｏｓ（ｒ）をすぐに導出することができ、そうでない場合、アルゴリズムは、ｔ＝１、２、．．．の場合、ｊがサンプリングされた行になり、Ｌ内に発見されるまでに、ｊ＝ＬＦ$^{t}$（ｒ）を判定する。この場合、ｐｏｓ（ｒ）＝ｐｏｓ（ｊ）＋ｔである。サンプリング方策は、Ｌ内の行が多くともμ回の繰返しで発見されること、および約（μ×ｏｃｃ）回の照会を介してパターンＰのｏｃｃ発生をランクデータ構造に位置付けることができることを確実にする。

【０２８５】

ｃｏｕｎｔ（Ｐ）は、ＦＭ索引によって対応される最後の基本演算ｅｘｔｒａｃｔ（ｉ，ｊ）を実施するように適合することができることに留意されたい。ｒを接尾辞Ｓ［ｊ，ｎ－１］を有するＭ’の行とし、ｒの値が既知であると仮定する。アルゴリズムは、Ｓ［ｊ］＝Ｆ［ｒ］を設定し、次いでｔ＝０、１、．．．、ｊ－ｉ－１の場合、ｓ［ｊ－１－ｔ］＝Ｌ［ＬＦ$^{t}$［ｒ］］を設定するサイクルを開始する。このサイクルの根底にある概念は、本発明者らが現在のシンボルのＬＦマッピング（ランクデータ構造を介して実施される）を繰返し算出し、したがってＳ［ｊ－１］から始まるＳ内で逆方向に飛び越すことである。ｊ－ｉ－１回のステップ後、Ｓ［ｉ］に到達したとき、プロセスは停止する。この手法は、ＢＷＴ転置で得られるものに類似しており、違いは、アレイＬＦが明示的に利用可能ではないが、そのエントリがランク演算を介してステップごとに生成されることに依拠する。この手法は、ＬＦアレイへの一定時間アクセスを依然として保証するが、ランクに対する圧縮手法が採用された場合、簡潔な空間記憶を保証する。

【０２８６】

上記で説明したアクセス、読取り、ランク、および探索のための化学的方法に加えて、「ｉｆ－ｔｈｅｎ－ｅｌｓｅ」演算を実施するための化学的方法も本明細書に記載される。「ｉｆ－ｔｈｅｎ－ｅｌｓｅ」演算は、指定のブール条件が真と評価されるか、それとも偽と評価されるかに応じて、異なる算出または行動を実行する命令、表現、または構成である。この演算を使用して、条件付きプログラムを書き込むことができる。各「ｉｆ」演算は、１つまたは複数の識別子の有無を試験し、その有無に応じて、「ｔｈｅｎ」または「ｅｌｓｅ」分岐へ進む。この演算は、複数の条件および対応する分岐を含むことができる。出力は、演算のすべての分岐から作り出すことができる。この手法により、複数の識別子ライブラリ内の識別子のすべて（たとえば、テラビット規模）を並列に演算することが可能になる。たとえば、ライブラリが数十億のデータオブジェクトを符号化する場合

、各オブジェクトを調べて出力を作り出す複素関数は、ＤＮＡに基づくプログラムとして設計することができ、ライブラリ上で並列に実行することができる。

【０２８７】

ビットのシフト、コピー、および移動に対する方策は、所望のプログラムの実行のために単一の識別子ライブラリを複数の入力識別子ライブラリに再配置することを可能にする。物理的に、１つの入力ライブラリおよび２つの出力ライブラリによる反応において、各ｉｆ－ｔｈｅｎ－ｅｌｓｅ演算を行うことができ、それらのライブラリ内のすべての識別子にわたって多重化することができる。流体伝達によって、１つの演算の出力ライブラリを別の演算の入力ライブラリへ向けることができる。ＤＮＡにおける各演算の実行は、従来のハードウェアと比較すると遅い可能性があるが、ＲＡＭおよび処理力によって制限される従来のハードウェアとは異なり、本明細書に記載するＤＮＡプラットホームは、大量の入力データオブジェクトにわたって同時に低電力でプログラムを実行することが可能である。

【０２８８】

図３３は、１つまたは複数のｉｆ－ｔｈｅｎ－ｅｌｓｅ演算を識別子のプール上で実行する方法３３００について説明する流れ図を示す。方法３３００のステップ３３０２は、１つまたは複数の入力シンボルストリングを表す識別子の第１のプールを取得することを伴う。ステップ３３０４は、ｉｆ－ｔｈｅｎ－ｅｌｓｅ演算を識別子上で実行して、識別子の第１のプールからの識別子のサブセットを有する中間プールを作成することを伴う。ステップ３３０６は、１つまたは複数のｉｆ－ｔｈｅｎ－ｅｌｓｅ演算を中間プール上で繰り返し、各ｉｆ－ｔｈｅｎ－ｅｌｓｅ演算後、識別子の最終プールが生成されるまで、新しい中間プールを作成することを伴い、最終プールは出力シンボルストリングを表す。

【０２８９】

プールは、粉末、液体、または固体の形態を有する。識別子のプール内の各識別子は、成分核酸分子を含む核酸分子である。成分核酸分子の少なくとも一部分は、ＰＣＲプライマーまたは親和性タグ付きオリゴヌクレオチドなどの１つまたは複数のプローブに個々に結合することが可能である。各識別子は、Ｍ個の層の各々とは別個の成分を含むことができ、ここで各層は、成分のセットを含む。

【０２９０】

ステップ３３０４および３３０６のｉｆ－ｔｈｅｎ－ｅｌｓｅ演算は、プローブを有する識別子の少なくとも１つの成分を標的とする。演算は、特有の成分を含むプール内の識別子にアクセスすることを伴うことができる。たとえば、プローブは、ＰＣＲプライマーであり、識別子は、ＰＣＲを介してアクセスされる。別の例として、プローブは、親和性タグ付きオリゴヌクレオチドであり、識別子は、親和性プルダウンアッセイを介してアクセスされる。複数のｉｆ－ｔｈｅｎ－ｅｌｓｅ演算を１つまたは複数のプール上で並列に実行することができる。たとえば、２つの演算が２つの識別子プール内で並列に実行される。

【０２９１】

方法３３００は、第１のプール、中間プール、または最終プールのうちの少なくとも１つを少なくとも２つの複製プールに分割することをさらに伴うことができる。この分割により、並列化されたｉｆ－ｔｈｅｎ－ｅｌｓｅ演算を可能にすることができる。十分な濃度の識別子を確実にするために、分割前に、たとえばＰＣＲを使用して、プールを複製することができる。２つまたはそれよりも多いプール（たとえば、中間プール）を組み合わせて、識別子の新しいプール（たとえば、新しい中間プールまたは第２のプール）を形成することができる。

【０２９２】

別の用途として、比較器回路のちょうど１つのインスタンス（約ｌｏｇ（ｎ））と同等のランタイムで、ＤＮＡ内の構造化されていないデータにわたって、長さｎのビットストリングの探索を実行することができる。従来のハードウェア上で実行される同等の探索は、データをＲＡＭへ移動させるコストおよび時間をさらに補償するはずである。さらにこ

の用途に対して、大きい格納データセットで一般的なように、データが符号化された後に考えられる任意のパターンに対して大きいデータセットを探索することを考慮する。たとえば、パターンはすべて、画像の大きいデータウェアハウスからの関心人物の写真である可能性がある。アーカイブが書き込まれた後、データから所望のパターンを発見することが可能な関数を考えることができる。そのような関数は、たとえば、機械学習アルゴリズムを使用して構築することができる。これらのパターン発見関数は、ブール回路表現を有しており、ｉｆ－ｔｈｅｎ－ｅｌｓｅ演算を使用してＤＮＡで実施することができる。演算は、学習済みモデルに従って、データセットのすべてのデータオブジェクトに同時に適用することができ、関心パターンに整合するオブジェクトを識別することができる。従来のハードウェア内で同じ演算を実行すると、ＲＡＭおよびプロセッサの利用可能性が制限されるため、高価で低速になるはずである。学習アルゴリズムが改善されると、更新プログラムを格納データセットに適用して、新しい関心情報を発見することができる。別の例示的な用途は、時空信号処理であり、特定のパターンを探索するとき、時間または空間の複数の小さい窓にわたって、畳込み関数（フーリエ変換などのデジタル信号処理関数の根底にある）を格納データに適用することができる。別の例として、安全性（たとえば、認証）または完全性（たとえば、安全性検査）の用途のために、大きいデータセットのオブジェクトにハッシュ関数を適用することができる。

【０２９３】

入力データがオブジェクトのセットを記述する場合、入力データは、識別子ライブラリ内で固有に符号化することができ、したがって各識別子が、セット内の可能なオブジェクトを表す。この符号化では、２つの識別子ライブラリ間の「ＡＮＤ」、「ＯＲ」、および「ＮＯＴ」演算の化学的方法が、当技術分野で定義されている「ＩＮＴＥＲＳＥＣＴＩＯＮ」、「ＵＮＩＯＮ」、および「ＣＯＭＰＬＥＭＥＮＴ」の集合演算を、一定のランタイムで実施する。集合包含（１つのセットが別のセットのサブセットであるかどうかを判定する）および均等物を実行するための化学的方法もまた、ＤＮＡプラットホーム上で実行することができる。これらの演算および関係をともに使用して、集合代数におけるあらゆる関数を実行することができる。

【０２９４】

ＤＮＡプラットホームはまた、機械学習（ＭＬ）に対する対応を含むことができる。ＭＬによる概念的な索引付けは、高次元のベクトルを利用して、データを組織化および注釈する。本明細書に記載するシステムは、そのようなデータに対して固有の効率的な記憶および照会を提供し、ＭＬモデルがデータ内の関連する概念および関係を捕捉する能力を完全に可能にすることができる。置換および合計などのＤＮＡ上の演算は、当技術分野では公知のように、概念的な関係を効率的に発見、記憶、および照会するために、ベクトル演算として実施することができる。識別子ライブラリをプラットホームとして使用して、記憶媒体に固有の記憶および構造を一体化する知的メモリを構築することができる。ＤＮＡに基づく記憶システムは、プラットホームに対するアルゴリズムおよびデータ構造に固有に対応する。データを概念的構造に継続的に組織化し、意味的に豊富な照会および推測を可能にするように、長い寿命の超低電力のメモリバンクを構成することができる。

【０２９５】

上記は、本開示の原理の単なる例示であり、記載の実装形態以外の形態で、装置および方法を実施することもでき、記載の実装形態は、限定ではなく例示の目的で提示されている。本開示を読めば、変形例および修正例が当業者には想到されよう。開示する特徴は、本明細書に記載する１つまたは複数の他の特徴とともに、任意の組合せおよび部分的組合せ（複数の依存する組合せおよび部分的組合せを含む）で実施することができる。上記で説明または例示した様々な特徴は、そのあらゆる構成要素を含めて、他のシステム内で組み合わせたり一体化したりすることができる。さらに、特定の特徴は、省略されてもよく、または実施されなくてもよい。

【０２９６】

変更、置換え、および修正の例は、当業者であれば確かめられるものであり、本明細書

に開示する情報の範囲を逸脱することなく加えることができる。本明細書に引用されるすべての参照は、参照により全体として組み込まれており、本出願の一部を構成する。

本発明は、例えば、以下を提供する。

（項目１）

核酸分子のプールに記憶されたデジタル情報からビットストリング内の特定のビットのランクを取得する方法であって、各ビットが、ビット値およびビット位置を有し、前記方法が、

（ａ）前記ビットストリングを表す識別子核酸分子の第１のプールを取得することであって、前記プールが、粉末、液体、または固体の形態を有し、前記第１のプール内の各識別子核酸分子が成分核酸分子を含み、前記成分核酸分子の少なくとも一部分が、１つまたは複数のプローブに結合するように構成される、第１のプールを取得することと、

（ｂ）前記ビットストリングから導出されたカウンタシンボルストリングを表す識別子核酸分子の第２のプールを取得することであって、各カウンタシンボルが、特有のビット値を有する前記ビットストリング内のすべてのｗ個のビットに対するビットの数のランニングカウントを示すｂ個のカウンタビットのストリングによって表される、第２のプールを取得することと、

（ｃ）（１）前記特定のビットに先行するｗ個のビットのすべてのブロック、または（２）前記特定のビットを含むｗ個のビットの前記ブロックを含む、前記特定のビットに先行するｗ個のビットのすべてのブロックのいずれかに対して、所与の値の前記ビットの数の前記ランニングカウントを示す対応するカウンタシンボルを表す前記第２のプール内の前記識別子核酸分子を少なくとも標的とするように、第２の一連のプローブを有する（ｂ）の前記第２のプールにアクセスすることによって、第１のカウントを取得することと、

（ｄ）（１）前記特定のビットに先行しもしくは前記特定のビットを含む、（ｃ）で計数されていないビットを表す、または（２）（ｃ）で計数されたが前記特定のビットに先行しないもしくは前記特定のビットを含まないビットを表す、前記第１のプール内の１つまたは複数の別個の識別子核酸分子を標的とするように、第１の一連のプローブを有する（ａ）の前記第１のプールにアクセスすることによって、第２のカウントを取得することと、

（ｅ）前記第１のカウントおよび前記第２のカウントから、前記ビットストリング内の前記特定のビットの前記ランクを取得することと

を含む方法。

（項目２）

前記第１のプール内の前記識別子核酸分子が、識別子の存在が、ビット位置の前記ビット値「１」を表すように前記ビットストリングを表す、項目１に記載の方法。

（項目３）

（ｃ）の前記第１のカウントが、前記特定のビットに先行するｗ個のビットのすべてのブロックを表すとき、（ｄ）の前記第１の一連のプローブが、前記特定のビットに先行するまたは前記特定のビットを含む、（ｃ）で計数されていないビットを表す前記第１のプール内の１つまたは複数の別個の識別子核酸分子を標的とし、前記ビットストリング内の前記特定のビットの前記ランクが、（ｅ）で前記第１および第２のカウントを合計することによって取得される、項目１および２のいずれかに記載の方法。

（項目４）

（ｃ）の前記第１のカウントが、前記特定のビットに先行し、前記特定のビットを含むｗ個のビットの前記ブロックを含む、ｗ個のビットのすべてのブロックを表すとき、前記第１の一連のプローブが、（ｃ）で計数されたが前記特定のビットに先行しないまたは前記特定のビットを含まないビットを表す前記第１のプール内の１つまたは複数の別個の識別子核酸分子を標的とし、前記ビットストリング内の前記特定のビットの前記ランクが、（ｅ）の前記第１のカウントから前記第２のカウントを引くことによって取得される、項目１および２のいずれかに記載の方法。

（項目５）

前記第１のカウントが、（ｃ）の前記標的とされた識別子核酸分子に対応するカウンタシンボル値を読み取ることによって取得される、項目１～４のいずれかに記載の方法。
（項目６）
前記第２のカウントが、（ｄ）からの前記標的とされた識別子核酸分子を読み取ることによって取得される、項目１～５のいずれかに記載の方法。
（項目７）
（ａ）の前記第１のプールが、（ｂ）の前記第２のプールである、項目１～６のいずれかに記載の方法。
（項目８）
（ａ）の前記第１のプールが、（ｂ）の前記第２のプールとは別個である、項目１～６のいずれかに記載の方法。
（項目９）
前記第１のプール内の対応する識別子核酸分子の存在が、前記ビット値１を示し、前記第１のプール内の対応する識別子核酸分子の不在が、前記ビット値０を示す、項目１～８のいずれかに記載の方法。
（項目１０）
前記ビットストリングが、長さｎを有し、ｂが$\log_2(n+1)$の上限である、項目１～９のいずれかに記載の方法。
（項目１１）
前記カウンタシンボルストリングが、ｎをｗで割った上限個のカウンタシンボルを含み、ｂにｎをｗで割った上限を掛けた値に対応する長さを有するカウンタビットストリングによって表される、項目１０に記載の方法。
（項目１２）
前記特定のビットが、ｗ個のビットの前記第１のブロックの範囲内である場合、ｗ個のビットの前記第１のブロックに先行する前記ランニングカウントが０である、項目１～１１のいずれかに記載の方法。
（項目１３）
前記特定のビットが、ｗ個のビットの前記第１のブロックの範囲内にない場合、前記特定のビットに先行するｗ個のビットのすべてのブロックの前記カウンタシンボルが、０からｗ＊Ｂ（ｘ）－１の位置範囲内の値１を有する前記ビットストリング内のビットの数を表し、０が前記ビットストリングの第１の位置であり、ｘが前記ビットストリング内の前記特定のビットの位置に対応し、Ｂ（ｘ）がｘをｗで割った下限である、項目１～１２のいずれかに記載の方法。
（項目１４）
（ｃ）の前記第２のプール内の前記標的とされた識別子核酸分子が、位置ｂ＊Ｂ（ｘ）およびｂ＊（Ｂ（ｘ）＋１）－１によって画定される前記範囲内であり、０の位置が、前記ビットストリング内の前記第１の位置に対応する、項目１３に記載の方法。
（項目１５）
前記第２のカウントが、前記位置の範囲ｗ＊Ｂ（ｘ）からｘ内に値１を有する前記ビットストリング内のビットの数に対応し、ｘが前記ビットストリング内の前記特定のビットの位置に対応し、位置０が、前記第１の位置であり、Ｂ（ｘ）が、ｘをｗで割った下限である、項目１～１４のいずれかに記載の方法。
（項目１６）
ｗがｂの値に設定される、項目１～１５のいずれかに記載の方法。
（項目１７）
ｗが１の値に設定される、項目１～１５のいずれかに記載の方法。
（項目１８）
前記第１のカウントが、前記特定のビットを含むｗ個のビットの前記ブロックに対応する前記カウンタシンボルを表す（ｃ）の識別子核酸分子を標的とすることによって取得され、前記ランクが前記第１のカウントと同等である、項目１７に記載の方法。

（項目１９）
ステップ（ｄ）が実行されない、項目１８に記載の方法。
（項目２０）
前記ビットストリング内のビットのブロックが、識別子核酸分子の前記第１のプール内の隣接して順序付けられた識別子核酸分子のブロックにマッピングされる、項目１～１９のいずれかに記載の方法。
（項目２１）
前記第１のプール内の識別子核酸分子の有無が、前記ビットストリング内のいずれのビット値とも直接相関しない、項目２０に記載の方法。
（項目２２）
前記ビットストリングの固定長のサブストリング（ワード）が、固定数の可能な固有の識別子核酸分子からの固定数の固有の識別子核酸分子を含むコードワードにマッピングされる、項目２０および２１のいずれかに記載の方法。
（項目２３）
追加の情報を使用して、前記第１および第２のプールからの識別子核酸分子の書込み、アクセス、および読取りのエラーを検出および補正することをさらに含む、項目１～２２のいずれかに記載の方法。
（項目２４）
前記追加の情報が、前記第１および第２のプールの前記識別子核酸分子に記憶される、項目２３に記載の方法。
（項目２５）
前記ビットストリングが、シンボルストリングを表し、前記ランクが、前記シンボルストリング内の特定のシンボルに対して取得される、項目１～２４のいずれかに記載の方法。
（項目２６）
前記シンボルストリング内の前記シンボルが、シンボル値のセットから選択され、（ｂ）の前記カウンタシンボルストリングが、前記特定のシンボル値を有するシンボルの数のランニングカウントを示す、項目２５に記載の方法。
（項目２７）
（ｂ）の識別子核酸分子の異なる第２のプールが、特有のシンボル値のインスタンスの数を計数する異なるカウンタシンボルストリングを表し、異なる各カウンタシンボルストリングが、対応する特有のシンボル値のインスタンスを計数する、項目２５および２６のいずれかに記載の方法。
（項目２８）
Ｍ個の選択された成分核酸分子を物理的に組み立てることによって、識別子核酸分子を形成することをさらに含み、前記Ｍ個の選択された成分核酸分子の各々が、Ｍ個の異なる層に分離された別個の成分核酸分子のセットから選択される、項目１～２７のいずれかに記載の方法。
（項目２９）
核酸分子のプールからデジタル情報をフェッチする方法であって、
（ａ）識別子核酸分子の第１のプールを取得することであって、前記プールが、粉末、液体、または固体の形態を有し、前記第１のプール内の各識別子核酸分子が成分核酸分子を含み、前記成分核酸分子の少なくとも一部分が、１つまたは複数のプローブに結合するように構成され、前記識別子核酸分子が、前記シンボル値が、前記第１のプール内の前記対応する識別子核酸分子の有無によって示されるようにシンボルストリングを表す、取得することと、
（ｂ）第１の一連のプローブを有する前記第１のプールにアクセスすることであり、前記第１の一連のプローブの各々が、前記成分核酸分子の少なくとも１つを標的として、前記第１のプールからの識別子核酸分子のサブセットを有する第２のプールを作成する、アクセスすることと、
（ｃ）前記第２のプールからの前記識別子核酸分子のサブセットの配列を読み取ること

と、
（ｄ）前記配列を使用して、（ａ）からの前記シンボルストリング内のシンボルの少なくともサブセットを取得することと
を含む方法。
（項目３０）
各識別子核酸分子が、Ｍ個の層の各々からの成分核酸配列を有する別個の成分核酸分子を含み、各層が、前記成分核酸配列のセットを含む、項目２９に記載の方法。
（項目３１）
前記Ｍ個の層が論理的に順序付けられる、項目３０に記載の方法。
（項目３２）
各層の前記成分核酸配列が論理的に順序付けられる、項目３１に記載の方法。
（項目３３）
前記識別子核酸分子が、前記識別子核酸配列を第１の層内の対応する成分核酸配列によって分類し、前記識別子核酸配列を第２の層内の対応する成分核酸配列によって細分し、残りのＭ－２個の層の各々に対して前記細分プロセスを繰り返すことによって、論理的に順序付けられた識別子核酸配列に対応する、項目３２に記載の方法。
（項目３４）
各識別子配列が、照会木内のパスとして表される一連の成分核酸配列を含み、前記照会木が、根ノードから開始し、各層に１つのインスタンスでＭ個のインスタンスにわたって分岐し、葉ノードで終了し、各葉ノードが、識別子核酸配列を表す、項目３３に記載の方法。
（項目３５）
前記第１の一連のプローブが、前記照会木内の前記根ノードからの部分パスまたはフルパスに対応する、項目３４に記載の方法。
（項目３６）
前記フルパスが、前記一連のプローブが、単一の識別子核酸分子を標的とするように、Ｍ個のプローブを含む根から葉までのパスに対応する、項目３５に記載の方法。
（項目３７）
前記部分パスが、前記一連のプローブが異なる配列を有する識別子核酸分子の複数の母集団を標的とするように、Ｍ個より少ないプローブに対応する、項目３６に記載の方法。
（項目３８）
異なる配列を有する識別子核酸分子の前記複数の母集団が、少なくとも前記第Ｍの層内の異なる成分核酸分子に対応する、項目３７に記載の方法。
（項目３９）
前記第１のプールが、複数の一連のプローブによってアクセスされる、項目２９～３８のいずれかに記載の方法。
（項目４０）
（ａ）の前記第１のプールを少なくとも２つの複製プールに分割することをさらに含み、（ｂ）、（ｃ）、および（ｄ）が、前記一連のプローブの各々を有する前記複製プールの各々で実行される、項目３９に記載の方法。
（項目４１）
前記少なくとも２つの複製プールに分割する前に、前記第１のプールを複製することをさらに含む、項目４０に記載の方法。
（項目４２）
プローブのサブシリーズを有する識別子核酸分子の第１のプールにアクセスして、識別子核酸分子の中間プールを作成することをさらに含む、項目４１に記載の方法。
（項目４３）
識別子核酸の中間プールを少なくとも２つの複製プールに分割することをさらに含む、項目４２に記載の方法。
（項目４４）

前記少なくとも２つの複製プールに分割する前に、前記中間プールを複製することをさらに含む、項目４３に記載の方法。
（項目４５）
後続のプローブのサブシリーズを有する識別子核酸分子の第１の中間プールにアクセスして、識別子核酸分子の第２の中間プールまたは識別子核酸分子の第２のプールを形成することをさらに含む、項目３９～４４のいずれかに記載の方法。
（項目４６）
識別子核酸分子の少なくとも２つの中間プールを組み合わせて、識別子核酸分子の別の中間プールまたは識別子核酸分子の第２のプールを形成することをさらに含む、項目３９～４５のいずれかに記載の方法。
（項目４７）
前記複製が、ポリメラーゼ連鎖反応（ＰＣＲ）によって実行される、項目４１および４４のいずれかに記載の方法。
（項目４８）
プローブがＰＣＲプライマーであり、アクセスがポリメラーゼ連鎖反応によって実行される、項目２９～４７のいずれかに記載の方法。
（項目４９）
前記プローブが親和性タグ付きオリゴヌクレオチドであり、アクセスが親和性プルダウンアッセイによって実行される、項目２９～４７のいずれかに記載の方法。
（項目５０）
長さｎのビットストリングを含むメッセージ内の長さｐの特定のビットパターンのカウントを取得する方法であって、
（ａ）ビットストリングＬを表す識別子核酸分子の第１のプールを取得することであって、前記ビットストリングＬが、前記メッセージのＢｕｒｒｏｗｓ－Ｗｈｅｅｌｅｒ変換行列の最後の列であり、前記第１のプールが、粉末、液体、または固体の形態を有し、前記第１のプール内の各識別子核酸分子が成分核酸分子を含み、前記成分核酸分子の少なくとも一部分が、１つまたは複数のプローブに結合するように構成される、第１のプールを取得することと、
（ｂ）前記ビットストリングＬから導出されたカウンタシンボルストリングを表す識別子核酸分子の第２のプールを取得することであって、各カウンタシンボルが、特有のビット値「１」を有する前記ビットストリングＬ内のすべてのｗ個のビットに対するビットの数のランニングカウントを示すｂ個のカウンタビットのストリングによって表される、第２のプールを取得することと、
（ｃ）一連のプローブを使用して、前記ビットストリングＬ内のビット値「１」の発生の総数に対する前記カウンタシンボルを表す前記第２のプールからの前記識別子核酸分子にアクセスすることと、
（ｄ）（ｃ）の前記アクセスされた識別子核酸分子を読み取って、前記ビットストリングＬ内のビット値「１」の発生の前記総数を計数することと、
（ｅ）（ｄ）からの各ビット値の発生の前記総数を使用して、前記メッセージの前記Ｂｕｒｒｏｗｓ－Ｗｈｅｅｌｅｒ変換行列の第１の列Ｆを再構築することと、
（ｆ）前記第１の列Ｆ内の前記第ｐのビット値の範囲を画定する第１の位置ｈおよび最後の位置ｚを、ｈおよびｚを含めて判定することと、
（ｇ）第１の一連のプローブを使用して、前記第１のプールおよび前記第２のプールからの前記識別子核酸分子にアクセスして、Ｌ内の位置ｈ－１における前記パターンの前記第（ｐ－ｉ）のビット値の前記ランク$r_{h-1}$を計算することであり、ｉ＝１である、計算することと、
（ｈ）第２の一連のプローブを使用して、前記第１のプールおよび第２のプールからの前記識別子核酸分子にアクセスして、Ｌ内の位置ｚにおける前記パターンの前記第（ｐ－ｉ）のビット値の前記ランク$r_{z}$を計算することと、
（ｉ）$r_{h-1}$が$r_{z}$に等しい場合、前記メッセージ内の前記パターンの発生の前記カウ

ントを０として設定することと、
（ｊ）そうではなく、$r_{h-1}$が$r_z$に等しくない場合、
（ｊ１）ｈをＦ内の前記第（ｐ－ｉ）のビット値の第（$r_{h-1}$＋１）のインスタンスの索引に設定し、
（ｊ２）ｚをＦ内の前記第（ｐ－ｉ）のビット値の第$r_z$のインスタンスの索引に設定し、
（ｊ３）ｉを１だけ増分し、
（ｊ４）ｉ＝ｐ－１になるまで、ステップ（ｇ）、（ｈ）、（ｉ）、（ｊ）、（ｊ１）、（ｊ２）、および（ｊ３）を複数回繰り返し、
（ｊ５）前記メッセージ内の前記パターンの発生の前記カウントをｚ－ｈ＋１として計算することとを含む、方法。
（項目５１）
（ａ）の前記第１のプールが、（ｂ）の前記第２のプールである、項目５０に記載の方法。
（項目５２）
（ａ）の前記第１のプールが、（ｂ）の前記第２のプールとは別個である、項目５０に記載の方法。
（項目５３）
前記第１のプール内の対応する識別子核酸分子の存在が、前記ビット値１を示し、前記第１のプール内の対応する識別子核酸分子の不在が、前記ビット値０を示す、項目５０～５２のいずれかに記載の方法。
（項目５４）
ｂが$log_2$（ｎ＋１）の上限である、項目５０～５３のいずれかに記載の方法。
（項目５５）
前記カウンタシンボルストリングが、ｎをｗで割った上限個のカウンタシンボルを含み、ｂにｎをｗで割った上限を掛けた値に対応する長さを有するカウンタビットストリングによって表される、項目５４に記載の方法。
（項目５６）
Ｌ内のｗ個のビットの前記第１のブロックに先行するあらゆるビット値の前記ランニングカウントが０である、項目５０～５５のいずれかに記載の方法。
（項目５７）
ｗがｂの値に設定される、項目５０～５６のいずれかに記載の方法。
（項目５８）
ｗが１の値に設定される、項目５０～５７のいずれかに記載の方法。
（項目５９）
前記ビットストリングＬ内のビットのブロックを、前記第１のプール内の隣接して順序付けられた識別子核酸分子のブロックにマッピングすることをさらに含む、項目５０～５８のいずれかに記載の方法。
（項目６０）
前記ビットストリングＬの固定長のサブストリングが、固有の識別子核酸分子の固定サイズのセットから選択された固定数の固有の識別子核酸分子によって表されるコードワードにマッピングされる、項目５９に記載の方法。
（項目６１）
前記第１および第２のプールからの識別子核酸分子の書込み、アクセス、および読取りのエラーを検出および補正するために、追加の情報が使用される、項目５０～６０のいずれかに記載の方法。
（項目６２）
前記追加の情報が、前記第１および第２のプールの前記識別子核酸分子に記憶される、項目６１に記載の方法。
（項目６３）

前記メッセージのＢｕｒｒｏｗｓ－Ｗｈｅｅｌｅｒ変換から導出された接尾辞アレイＳＡを表す識別子核酸分子の第３のプールを取得することをさらに含み、ＳＡの各要素が、前記メッセージ内のＬの対応する要素の位置を示す少なくともｌｏｇ$_2$（ｎ）個のビットのビットストリングによって表される、項目５０～６２のいずれかに記載の方法。

（項目６４）

前記カウントが０より大きいとき、ｈおよびｚに対する最終値を含むｈとｚとの間の位置における前記接尾辞アレイ内の要素に対応する前記第３のプール内の前記識別子核酸分子にアクセスすることによって、前記メッセージ内の前記パターンの前記発生を位置付けることをさらに含む、項目６３に記載の方法。

（項目６５）

前記メッセージを表す識別子核酸分子の第４のプールを取得することをさらに含む、項目６４に記載の方法。

（項目６６）

前記第１の場所および前記第１の場所を取り囲む位置の近傍に対応する前記第４のプール内の前記識別子核酸分子にアクセスすることによって、前記パターンの前記第１の場所のコンテキストを抽出することをさらに含む、項目６５に記載の方法。

（項目６７）

長さｎのビットストリングを含むメッセージ内の長さｐの特定のビットパターンのカウントを取得する方法であって、

（ａ）ビットストリングＬを表す識別子核酸分子の第１のプールを取得することであって、前記ビットストリングＬが、前記メッセージのＢｕｒｒｏｗｓ－Ｗｈｅｅｌｅｒ変換行列の最後の列であり、前記第１のプールが、粉末、液体、または固体の形態を有し、前記第１のプール内の各識別子核酸分子が成分核酸分子を含み、前記成分核酸分子の少なくとも一部分が１つまたは複数のプローブに結合するように構成される、第１のプールを取得することと、

（ｂ）特有のビット値を有するビットの数のランニングカウントを表す、前記ビットストリングＬから導出されたカウンタシンボルストリングを表す識別子核酸分子の第２のプールを取得することと、

（ｃ）前記第１のプールおよび前記第２のプールからの識別子核酸分子に選択的にアクセスすることによって、前記メッセージ内の前記特定のビットパターンの前記カウントを取得することと

を含む方法。

（項目６８）

ステップ（ｃ）が、前記Ｂｕｒｒｏｗｓ－Ｗｈｅｅｌｅｒ変換行列の第１の列Ｆを再構築することをさらに含む、項目６７に記載の方法。

（項目６９）

ステップ（ｃ）が、一連のプローブを使用して、前記ビットストリングＬ内の前記特有のビット値の発生の総数に対する前記カウンタシンボルを表す前記第２のプールからの前記識別子核酸分子にアクセスすることと、前記特有のビット値の発生の前記総数を使用して、Ｆを再構築することとを含む、項目６８に記載の方法。

（項目７０）

ステップ（ｃ）が、前記パターン内の第ｐのビット値を有するＦ内の位置の範囲を判定することをさらに含む、項目６８および６９のいずれかに記載の方法。

（項目７１）

Ｆ内の位置の前記範囲が、第１の位置ｈおよび最後の位置ｚによって、ｈおよびｚを含めて画定される、項目７０に記載の方法。

（項目７２）

ステップ（ｃ）が、前記第ｐのビット後の前記パターン内の先行する各ビットに対して、

一連のプローブを使用して、前記第１のプールおよび前記第２のプールからの前記識別子核酸分子にアクセスして、前記範囲にすぐ先行する位置におけるＬ内の前記対応するビ

ット値の第１のランク、および前記範囲の末端の位置におけるＬ内の前記対応するビット値の第２のランクを判定することと、
前記第１のランクおよび前記第２のランクを使用して、前記範囲を、前記パターン内の後続のビットに先行する前記対応するビットのインスタンスを有するＦ内の位置の範囲に更新することとをさらに含む、項目７０および７１のいずれかに記載の方法。
（項目７３）
前記第１のランクが、Ｌ内の位置ｈ－１における前記パターン内のそれぞれの先行するビット値であり、前記第２のランクが、Ｌ内の位置ｚにおける前記パターン内のそれぞれの先行するビット値である、項目７２に記載の方法。
（項目７４）
前記範囲を更新することが、前記第１のランクに基づいて、ｈをＦ内の前記パターン内の前記それぞれの先行するビット値の１つのインスタンスの位置に設定することを含む、項目７３に記載の方法。
（項目７５）
前記範囲を更新することが、前記第２のランクに基づいて、ｚをＦ内の前記パターン内の前記それぞれの先行するビット値の１つのインスタンスの位置に設定することを含む、項目７３および７４のいずれかに記載の方法。
（項目７６）
ステップ（ｃ）が、前記第１および第２のランクの前記最終値に基づいて、前記メッセージ内の前記パターンの発生の前記カウントを設定することをさらに含む、項目７２～７５のいずれかに記載の方法。
（項目７７）
発生の前記カウントが、前記第１および第２のランクの前記最終値間の差である、項目７６に記載の方法。
（項目７８）
前記第１および第２のランクが、前記第ｐのビットまたはあらゆる先行するビットに対して互いに等しい場合、発生の前記カウントが０に設定される、項目７６に記載の方法。
（項目７９）
長さ$n_s$のシンボルストリングを含むメッセージ内の長さ$p_s$の特定のシンボルパターンのカウントを取得する方法であって、各シンボルが、ｒ個のシンボル値のセットから選択され、前記方法が、
（ａ）前記メッセージのＢｕｒｒｏｗｓ－Ｗｈｅｅｌｅｒ変換行列の最後の列であるシンボルストリングＬを表す識別子核酸分子の第１のプールを取得することであって、前記第１のプールが、粉末、液体、または固体の形態を有し、前記第１のプール内の各識別子核酸分子が成分核酸分子を含み、前記成分核酸分子の少なくとも一部分が、１つまたは複数のプローブに結合するように構成される、第１のプールを取得することと、
（ｂ）識別子核酸分子のｒ個の第２のプールを取得することであって、前記識別子核酸分子の各々が、Ｌから導出されたカウンタシンボルストリング$C_v$に対応し、ここでｖ＝１、２、．．．、ｒであり、対応するシンボル値$R_v$を有するＬ内のシンボルの数のランニングカウントを表す、ｒ個の第２のプールを取得することと、
（ｃ）前記第１のプールおよび前記ｒ個の第２のプールからの識別子核酸分子に選択的にアクセスすることによって、前記メッセージ内の長さ$p_s$の特定のシンボルパターンの前記カウントを取得することとを含む方法。
（項目８０）
ステップ（ｃ）が、前記Ｂｕｒｒｏｗｓ－Ｗｈｅｅｌｅｒ変換行列の第１の列Ｆを再構築することをさらに含む、項目７９に記載の方法。
（項目８１）
ステップ（ｃ）が、一連のプローブを使用して、Ｌ内の対応する各シンボル値$R_v$の発生の総数を表す前記ｒ個の第２のプールの各々における前記最後のカウンタシンボルからの前記識別子核酸分子にアクセスすることと、対応する各シンボル値$R_v$の発生の前記総

数を使用して、Ｆを再構築することとを含む、項目８０に記載の方法。
（項目８２）
ステップ（ｃ）が、前記パターン内の第ｐのシンボル値を有するＦ内の位置の範囲を判定することをさらに含む、項目８０および８１のいずれかに記載の方法。
（項目８３）
Ｆ内の位置の前記範囲が、第１の位置ｈおよび最後の位置ｚによって、ｈおよびｚを含めて画定される、項目８２に記載の方法。
（項目８４）
ステップ（ｃ）が、前記第ｐのシンボル後の前記パターン内の先行する各シンボルに対して、
一連のプローブを使用して、前記第１のプールおよび前記対応する第２のプールからの前記識別子核酸分子にアクセスして、前記範囲にすぐ先行する位置におけるＬ内の前記対応するシンボル値の第１のランク、および前記範囲の末端の位置におけるＬ内の前記対応するシンボル値の第２のランクを判定することと、
前記第１のランクおよび前記第２のランクを使用して、前記範囲を、前記パターン内の後続のシンボルに先行する前記対応するシンボルのインスタンスを有するＦ内の位置の範囲に更新することと
をさらに含む、項目８２および８３のいずれかに記載の方法。
（項目８５）
前記第１のランク$r_{h-1}$が、Ｌ内の位置ｈ－１における前記パターン内のそれぞれの先行するシンボル値であり、前記第２のランク$r_z$が、Ｌ内の位置ｚにおける前記パターン内のそれぞれの先行するシンボル値である、項目８４に記載の方法。
（項目８６）
前記範囲を更新することが、ｈをＦ内の前記パターン内の前記それぞれの先行するシンボル値の第（$r_{h-1}$＋１）のインスタンスの位置に設定することを含む、項目８５に記載の方法。
（項目８７）
前記範囲を更新することが、ｚをＦ内の前記パターン内の前記それぞれの先行するシンボル値の第$r_z$のインスタンスの索引に設定することを含む、項目８５および８６のいずれかに記載の方法。
（項目８８）
ステップ（ｃ）が、前記第１および第２のランクの前記最終値に基づいて、前記メッセージ内の前記パターンの発生の前記カウントを設定することをさらに含む、項目８４～８７のいずれかに記載の方法。
（項目８９）
発生の前記カウントが、前記第１および第２のランクの前記最終値間の差である、項目８８に記載の方法。
（項目９０）
前記第１および第２のランクが、前記第ｐのシンボルまたはあらゆる先行するシンボルに対して互いに等しい場合、発生の前記カウントが０に設定される、項目８８に記載の方法。
（項目９１）
識別子核酸分子の前記第１のプールが、ｒ個の第１のプールのうちの１つであり、前記ｒ個の第１のプールの各々が、ビットストリング$L_v$に対応し、ここでｖ＝１、２、．．．、ｒであり、したがって$L_v$の要素が、前記シンボル値$R_v$に一致するＬの要素に対してビット値「１」、そうでない場合はビット値「０」を有し、または逆も同様である、項目７９～９０のいずれかに記載の方法。
（項目９２）
$L_v$に対応する前記第１のプールが、前記パターン内のシンボル値$R_v$の前記第１および第２のランクを判定するために使用される、項目９１に記載の方法。

（項目９３）
前記メッセージの前記Ｂｕｒｒｏｗｓ－Ｗｈｅｅｌｅｒ変換から導出された接尾辞アレイＳＡを表す識別子核酸分子のプールであるＳＡプールを取得することをさらに含み、ＳＡの各要素が、前記メッセージ内のＬの対応する要素の位置を示す少なくともｌｏｇ$_2$（ｎ）個のビットのビットストリングによって表される、項目７９～９２のいずれかに記載の方法。
（項目９４）
前記カウントが０より大きいと仮定して、Ｆ内の位置の最終範囲によって与えられる位置における前記ＳＡの要素に対応する前記ＳＡプール内の識別子核酸分子にアクセスすることによって、前記メッセージ内の前記パターンの前記発生を位置付けることをさらに含む、項目９３に記載の方法。
（項目９５）
前記メッセージを表す識別子核酸分子のメッセージプールを取得することをさらに含む、項目９４に記載の方法。
（項目９６）
第１の場所および前記第１の場所を取り囲む位置の近傍に対応する前記メッセージプール内の前記識別子核酸分子にアクセスすることによって、前記パターンの前記第１の場所のコンテキストを抽出することをさらに含む、項目９５に記載の方法。
（項目９７）
ステップ（ｃ）が、
（ｃ．１）前記第１の列Ｆ内の前記第ｐのシンボル値の範囲を画定する第１の位置ｈおよび最後の位置ｚを、ｈおよびｚを含めて判定することと、
（ｃ．２）一連のプローブを使用して、第１のプールおよび第２のプールからの前記識別子核酸分子にアクセスし、Ｌ内の位置ｈ－１における前記パターンの第（ｐ－ｉ）のシンボル値のランク$r_{h-1}$を計算することであり、ｉ＝１である、計算することと、
（ｃ．３）一連のプローブを使用して、第１のプールおよび第２のプールからの前記識別子核酸分子にアクセスし、Ｌ内の位置ｚにおける前記パターンの前記第（ｐ－ｉ）のシンボル値のランク$r_z$を計算することと、
（ｃ．４）$r_{h-1}$が$r_z$に等しい場合、前記メッセージ内の前記パターンの発生の前記カウントを０として設定することと、
（ｃ．５）そうではなく、$r_{h-1}$が$r_z$に等しくない場合、
（ｃ．５．Ａ）ｈをＦ内の前記第（ｐ－ｉ）のシンボル値の第（$r_{h-1}$＋１）のインスタンスの索引に設定し、
（ｃ．５．Ｂ）ｚをＦ内の前記第（ｐ－ｉ）のシンボル値の第$r_z$のインスタンスの索引に設定し、
（ｃ．５．Ｃ）ｉを１だけ増分し、
（ｃ．５．Ｄ）ｉ＝ｐ－１になるまで、ステップ（ｃ．２）、（ｃ．３）、（ｃ．４）、（ｃ．５）、（ｃ．５．Ａ）、（ｃ．５．Ｂ）、および（ｃ．５．Ｃ）を複数回繰り返し、
（ｃ．５．Ｅ）前記メッセージ内の前記パターンの発生の前記カウントをｚ－ｈ＋１として計算することと
を含む、項目７９～９６のいずれかに記載の方法。
（項目９８）
（ｃ．２）の前記第１のプールおよび前記第２のプールが、前記パターンの前記第（ｐ－ｉ）のシンボル値に対応する、項目９７に記載の方法。
（項目９９）
（ｃ．３）の前記第１のプールおよび前記第２のプールが、前記パターンの前記第（ｐ－ｉ）のシンボル値に対応する、項目９７および９８のいずれかに記載の方法。
（項目１００）
核酸分子内へデジタル情報を記憶する方法であって、

複数のブロックを取得することであって、各ブロックがシンボルストリングを含み、ブロックＩＤに関連付けられる、取得することと、
前記複数のブロックのうちの１つのブロックをコンテナに割り当てることと、
前記ブロックを、前記コンテナに関連付けられる複数の識別子核酸配列にマッピングすることであって、各識別子核酸配列が成分核酸配列を含み、前記成分核酸配列の少なくとも一部分が、１つまたは複数のプローブに結合するように構成される、マッピングすることと、
前記複数の識別子核酸配列の個々の識別子核酸分子を構築することと、
前記割り当てられたコンテナ内に前記個々の識別子核酸分子を記憶することと
を含み、前記コンテナおよび前記コンテナに関連付けられた前記複数の識別子核酸配列の識別情報を含む物理アドレスが、前記関連付けられたブロックＩＤを使用して判定されるように構成される、方法。
（項目１０１）
前記ブロックＩＤが、整数、ストリング、トリプル、属性リスト、または意味注釈である、項目１００に記載の方法。
（項目１０２）
前記物理アドレスが、前記関連付けられたブロックＩＤを使用して前記物理アドレスにアクセスすることを容易にするように設計されたデータ構造に記憶される、項目１００および１０１のいずれかに記載の方法。
（項目１０３）
前記データ構造が、Ｂ木、トライ、またはアレイのうちの１つである、項目１００～１０２のいずれかに記載の方法。
（項目１０４）
前記データ構造の少なくとも一部分が、前記デジタル情報とともに索引内に記憶される、項目１０２および１０３のいずれかに記載の方法。
（項目１０５）
前記索引が、第２のコンテナに関連付けられた第２の複数の識別子核酸配列を含む、項目１０４に記載の方法。
（項目１０６）
前記索引が、Ｂ木データ構造を含み、前記Ｂ木の各ノードが、前記第２の複数の識別子核酸配列の別個の複数の識別子核酸分子を含む、項目１０５に記載の方法。
（項目１０７）
前記Ｂ木内の前記ブロックＩＤを探索することが、
第１のノードを含む前記別個の複数の識別子核酸分子にアクセスすることと、
前記第１のノードの値を読み取ることと、
ステップ（ｉ）および（ｉｉ）のプロセスを後続のノードに対して繰り返すことと
を含み、前記後続のノードを含む前記別個の複数の識別子核酸分子の識別情報が、前記第１のノードの前記値に関連する前記ブロックＩＤによって判定される、項目１０６に記載の方法。
（項目１０８）
前記第１のノードが、前記Ｂ木の根ノードであり、ステップ（ｉ）および（ｉｉ）の前記プロセスが、前記Ｂ木の葉ノードの値が読み取られるまで継続し、前記葉ノードの前記値が、前記ブロックＩＤに対するブロックが存在するかどうかを通信し、前記ブロックＩＤが存在する場合、前記ブロックの前記物理アドレスを通信するように構成される、項目１０７に記載の方法。
（項目１０９）
前記索引が、トライデータ構造を含み、前記トライの各ノードが、前記第２の複数の識別子核酸配列の別個の複数の識別子核酸分子を含む、項目１０５に記載の方法。
（項目１１０）
前記ブロックＩＤが、シンボルストリングであり、前記トライデータ構造内の各ノード

が、前記シンボルストリングの可能な接頭辞に対応する、項目１０９に記載の方法。
（項目１１１）
前記トライデータ構造の葉ノードが、前記葉ノード内の前記トライデータ構造によって指定された前記シンボルストリングに一致する前記ブロックＩＤに関連付けられた前記物理アドレスを表す、項目１１０に記載の方法。
（項目１１２）
前記データ構造が、アレイであり、前記アレイの各要素が、前記第２の複数の識別子核酸配列の別個の複数の識別子核酸分子を含む、項目１０４に記載の方法。
（項目１１３）
前記アレイ内の各要素が、ブロックＩＤに対応する、項目１１２に記載の方法。
（項目１１４）
前記アレイの各要素が、前記関連付けられたブロックＩＤの前記物理アドレスを記憶する、項目１１３に記載の方法。
（項目１１５）
前記第２の複数の識別子核酸配列を含む識別子核酸分子のプールから、一連のプローブを使用して、物理アドレスにアクセスすることをさらに含む、項目１０４～１１４のいずれかに記載の方法。
（項目１１６）
前記データ構造が、磁気記憶デバイス、光記憶デバイス、フラッシュメモリデバイス、またはクラウドストレージに記憶される、項目１０２に記載の方法。
（項目１１７）
前記物理アドレスが、前記物理アドレスを追加のデータ構造に記憶することなく、前記ブロックＩＤが前記物理アドレスにマッピングされるように、前記ブロックＩＤに固有に構成される、項目１０１に記載の方法。
（項目１１８）
前記ブロックＩＤが、前記物理アドレスに関連付けられた前記複数の識別子核酸配列のうちのすべての識別子核酸配列によって共用される複数の成分核酸配列にマッピングされる、項目１１７に記載の方法。
（項目１１９）
ブロックに関連付けられた複数の識別子核酸配列が、隣接して順序付けられた識別子核酸配列を含み、したがって前記複数の識別子核酸配列が、前記識別子範囲内の前記第１および最後の識別子核酸分子の識別情報を含む識別子範囲によって、対応する物理アドレスに指定される、項目１００～１１８のいずれかに記載の方法。
（項目１２０）
前記識別子範囲内の前記第１および最後の識別子核酸配列が、整数によって表される、項目１１９に記載の方法。
（項目１２１）
一連のプローブを使用することによって、ブロックに関連付けられた複数の核酸分子にアクセスすることをさらに含む、項目１００～１２０のいずれかに記載の方法。
（項目１２２）
前記プローブがＰＣＲプライマーであり、前記アクセスが、ポリメラーゼ連鎖反応によって実行される、項目１１５および１２１のいずれかに記載の方法。
（項目１２３）
前記プローブが親和性タグ付きオリゴヌクレオチドであり、前記アクセスが、親和性プルダウンアッセイによって実行される、項目１１２および１２１のいずれかに記載の方法。
（項目１２４）
ブロックＩＤが位置である、項目１００～１２３のいずれかに記載の方法。
（項目１２５）
前記位置が、親シンボルストリングの対応するブロックによって表されるシンボルストリングの位置である、項目１２４のいずれかに記載の方法。

（項目１２６）
前記親シンボルストリングが、別のシンボルストリングにおけるパターンの前記発生の計数または位置付けのためのデータ構造を含む、項目１２５に記載の方法。
（項目１２７）
前記データ構造が、カウンタアレイ、Ｂｕｒｒｏｗｓ－Ｗｈｅｅｌｅｒ変換（ＢＷＴ）行列の列、接尾辞アレイ、接尾辞木、または転置索引のうちの１つである、項目１２６に記載の方法。
（項目１２８）
核酸分子に記憶されたデジタル情報を演算する方法であって、
識別子核酸分子の第１のプールを取得することであって、前記プールが、粉末、液体、または固体の形態を有し、前記第１のプール内の各識別子核酸分子が成分核酸分子を含み、前記成分核酸分子の少なくとも一部分が、１つまたは複数のプローブに結合するように構成され、前記識別子核酸分子が、入力シンボルストリングを表す、取得することと、
前記第１のプール内の前記識別子核酸分子でｉｆ－ｔｈｅｎ－ｅｌｓｅ演算を実行することであって、前記ｉｆ－ｔｈｅｎ－ｅｌｓｅ演算が、プローブを有する前記成分核酸分子のうちの少なくとも１つを標的とし、前記第１のプールからの識別子核酸分子のサブセットを有する中間プールを作成する、実行することと、
ステップ（ｂ）を繰り返すこと
を含み、出力シンボルストリングの少なくとも一部分を表す識別子核酸分子の最終プールが作成されるまで、すべての後続のステップで、前記中間プールが前記第１のプールに取って代わる、方法。
（項目１２９）
前記第１のプール内の各識別子核酸分子が、Ｍ個の層の各々から別個の成分核酸分子を含み、各層が、成分核酸分子のセットを含む、項目１２８に記載の方法。
（項目１３０）
ｉｆ－ｔｈｅｎ－ｅｌｓｅ演算が、特有の成分核酸分子を含むプール内の識別子核酸分子にアクセスすることを含む、項目１２８および１２９のいずれかに記載の方法。
（項目１３１）
プローブがＰＣＲプライマーであり、アクセスが、ポリメラーゼ連鎖反応によって実行される、項目１３０のいずれかに記載の方法。
（項目１３２）
プローブが親和性タグ付きオリゴヌクレオチドであり、アクセスが、親和性プルダウンアッセイによって実行される、項目１３０に記載の方法。
（項目１３３）
識別子核酸分子の１つまたは複数のプールで並列に、２つまたはそれよりも多いｉｆ－ｔｈｅｎ－ｅｌｓｅ演算が実行される、項目１２８～１３２のいずれかに記載の方法。
（項目１３４）
前記第１のプール、前記中間プール、または前記最終プールのうちの少なくとも１つを少なくとも２つの複製プールに分割することをさらに含む、項目１２８～１３３に記載の方法。
（項目１３５）
前記第１のプール、前記中間プール、または前記最終プールのうちの前記少なくとも１つを分割前に複製することをさらに含む、項目１３４に記載の方法。
（項目１３６）
前記複製が、ポリメラーゼ連鎖反応（ＰＣＲ）によって実行される、項目１３５に記載の方法。
（項目１３７）
識別子核酸分子の少なくとも２つの中間プールを組み合わせて、識別子核酸分子の新しい中間プールまたは識別子核酸分子の第２のプールを形成することをさらに含む、項目１２８～１３６のいずれかに記載の方法。

（項目１３８）
核酸分子のプールに記憶されたデジタル情報からシンボルストリング内の特定の位置における特定のシンボル値のランクを取得する方法であって、各シンボルが、シンボル値およびシンボル位置を有し、前記方法が、
（ａ）前記シンボルストリングを表す識別子核酸分子の第１のプールを取得することであって、前記プールが、粉末、液体、または固体の形態を有し、前記第１のプール内の各識別子核酸分子が成分核酸分子を含み、前記成分核酸分子の少なくとも一部分が、１つまたは複数のプローブに結合するように構成される、第１のプールを取得することと、
（ｂ）前記シンボルストリングから導出されたカウンタシンボルストリングを表す識別子核酸分子の第２のプールを取得することであって、各カウンタシンボルが、前記シンボルストリングのすべてのｗ個のシンボル内の前記特定のシンボル値のランニングカウントを表す、第２のプールを取得することと、
（ｃ）（１）前記特定の位置に先行するｗ個のシンボルのすべてのブロック、または（２）前記特定の位置を含むｗ個のシンボルのブロックを含む、前記特定の位置に先行するｗ個のシンボルのすべてのブロックのいずれかに対して、前記特定のシンボル値の前記ランニングカウントを示す対応するカウンタシンボルを表す前記第２のプール内の前記識別子核酸分子を少なくとも標的とするように、第２の一連のプローブを有する（ｂ）の前記第２のプールにアクセスすることによって、第１のカウントを取得することと、
（ｄ）（１）前記特定の位置に先行しもしくは前記特定の位置を含む、（ｃ）で計数されていないシンボルを表す、または（２）（ｃ）で計数されたが前記特定の位置に先行しないもしくは前記特定の位置を含まないシンボルを表す、前記第１のプール内の１つまたは複数の別個の識別子核酸分子を標的とするように、第１の一連のプローブを有する（ａ）の前記第１のプールにアクセスすることによって、第２のカウントを取得することと、
（ｅ）前記第１のカウントおよび前記第２のカウントから、前記シンボルストリング内の前記特定の位置における前記特定のシンボル値の前記ランクを取得することと
を含む方法。
（項目１３９）
前記第１のプール内の前記識別子核酸分子が、識別子核酸分子の存在が、前記シンボル位置における特定のシンボル値を表すように、前記シンボルストリングにマッピングされたビットストリングを表す、項目１３８に記載の方法。
（項目１４０）
（ｃ）の前記第１のカウントが、前記特定の位置に先行するｗ個のシンボルのすべてのブロックを表すとき、（ｄ）の前記第１の一連のプローブが、前記特定の位置に先行しまたは前記特定の位置を含む、（ｃ）で計数されていないシンボルを表す前記第１のプール内の１つまたは複数の別個の識別子核酸分子を標的とし、前記シンボルストリング内の前記特定の位置における前記特定のシンボル値の前記ランクが、（ｅ）で前記第１および第２のカウントを合計することによって取得される、項目１３８および１３９のいずれかに記載の方法。
（項目１４１）
（ｃ）の前記第１のカウントが、前記特定の位置を含むｗ個のシンボルの前記ブロックを含む、前記特定の位置に先行するｗ個のシンボルのすべてのブロックを表すとき、前記第１の一連のプローブが、（ｃ）で計数されたが前記特定の位置に先行しないまたは前記特定の位置を含まないシンボルを表す前記第１のプール内の１つまたは複数の別個の識別子核酸分子を標的とし、前記シンボルストリング内の前記特定の位置における前記特定のシンボル値の前記ランクが、（ｅ）で前記第１のカウントから前記第２のカウントを引くことによって取得される、項目１３８および１３９のいずれかに記載の方法。
（項目１４２）
前記第１のカウントが、（ｃ）の前記標的とされた識別子核酸分子に対応するカウンタシンボル値を読み取ることによって取得される、項目１３８～１４１のいずれかに記載の方法。

（項目１４３）
前記第２のカウントが、（ｄ）からの前記標的とされた識別子核酸分子を読み取ることによって取得される、項目１３８～１４２のいずれかに記載の方法。
（項目１４４）
（ａ）の前記第１のプールが、（ｂ）の前記第２のプールである、項目１３８～１４３のいずれかに記載の方法。
（項目１４５）
（ａ）の前記第１のプールが、（ｂ）の前記第２のプールとは別個である、項目１３８～１４３のいずれかに記載の方法。
（項目１４６）
前記第１のプール内の対応する識別子核酸分子の存在が、第１のシンボル値を示し、前記第１のプール内の対応する識別子核酸分子の不在が、第２のシンボル値を示す、項目１３８～１４５のいずれかに記載の方法。
（項目１４７）
前記シンボルストリングが、長さｎを有し、前記カウンタシンボルが、ｂビットによって表され、ここでｂが$log_2$（ｎ＋１）の上限である、項目１３８～１４６のいずれかに記載の方法。
（項目１４８）
前記カウンタシンボルストリングが、ｎをｗで割った上限個のカウンタシンボルを含み、ｂにｎをｗで割った上限を掛けた値に対応する長さを有するカウンタビットストリングによって表される、項目１４７に記載の方法。
（項目１４９）
前記特定の位置が、ｗ個のシンボルの前記第１のブロックの範囲内である場合、ｗ個のシンボルの前記第１のブロックに先行する前記ランニングカウントが０である、項目１３８～１４８のいずれかに記載の方法。
（項目１５０）
前記特定の位置が、ｗ個のシンボルの前記第１のブロックの範囲内にない場合、前記特定の位置に先行するｗ個のシンボルのすべてのブロックの前記カウンタシンボルが、前記シンボルストリングの０からｗ＊Ｂ（ｘ）－１の位置範囲内の前記特定のシンボル値の発生の数を表し、０が前記シンボルストリングの第１の位置であり、ｘが前記シンボルストリング内の前記特定の位置に対応し、Ｂ（ｘ）がｘをｗで割った下限である、項目１３８～１４９のいずれかに記載の方法。
（項目１５１）
（ｃ）の前記第２のプール内の前記標的とされた識別子核酸分子が、位置ｂ＊Ｂ（ｘ）およびｂ＊（Ｂ（ｘ）＋１）－１によって画定される前記範囲内であり、０の位置が、前記シンボルストリング内の前記第１の位置に対応する、項目１５０に記載の方法。
（項目１５２）
前記第２のカウントが、前記シンボルストリングのｗ＊Ｂ（ｘ）からｘの前記位置範囲内の前記特定のシンボル値の発生の数に対応し、ｘが前記シンボルストリング内の前記特定の位置に対応し、位置０が、前記第１の位置であり、Ｂ（ｘ）が、ｘをｗで割った下限である、項目１３８～１５１のいずれかに記載の方法。
（項目１５３）
ｗが、前記シンボルストリングのｗ個のシンボルを表すビットの長さが、ビットの長さｂと同等になり、前記カウンタシンボルを表すように設定される、項目１３８～１５２のいずれかに記載の方法。
（項目１５４）
ｗが１の値に設定される、項目１３８～１５２のいずれかに記載の方法。
（項目１５５）
前記第１のカウントが、前記特定の位置を含むｗ個のシンボルの前記ブロックに対応する前記カウンタシンボルを表す（ｃ）の識別子核酸分子を標的とすることによって取得さ

れ、前記ランクが前記第１のカウントと同等である、項目１５４に記載の方法。
（項目１５６）
ステップ（ｄ）が実行されない、項目１５５に記載の方法。
（項目１５７）
前記シンボルストリング内のｗ個のシンボルのブロックが、識別子核酸分子の前記第１のプール内の隣接して順序付けられた識別子核酸分子のブロックにマッピングされる、項目１３８～１５６のいずれかに記載の方法。
（項目１５８）
前記シンボルストリング内の前記シンボルがビットであり、各ビットが識別子核酸分子にマッピングされ、したがって識別子核酸分子の前記第１のプール内の前記識別子の有無が、前記ビットの値を示す、項目１５７に記載の方法。
（項目１５９）
前記シンボルストリングの固定長のサブストリングが、固定数の可能な固有の識別子核酸分子からの固定数の固有の識別子核酸分子を含むコードワードにマッピングされる、項目１５７および１５８のいずれかに記載の方法。
（項目１６０）
追加の情報を使用して、前記第１および第２のプールからの識別子核酸分子の書込み、アクセス、および読取りのエラーを検出および補正することをさらに含む、項目１３８～１５９のいずれかに記載の方法。
（項目１６１）
前記追加の情報が、前記第１および第２のプールの前記識別子核酸分子に記憶される、項目１６０に記載の方法。
（項目１６２）
前記シンボルストリングが、ビットストリングを表す、項目１３８～１６１のいずれかに記載の方法。
（項目１６３）
前記シンボルストリングの各シンボルが、固定数のビットに対応する、項目１６２に記載の方法。
（項目１６４）
（ｂ）の識別子核酸分子の異なる第２のプールが、特有のシンボル値のインスタンスの数を計数する異なるカウンタシンボルストリングを表し、異なる各カウンタシンボルストリングが、対応する特有のシンボル値のインスタンスを計数する、項目１６２および１６３のいずれかに記載の方法。
（項目１６５）
Ｍ個の選択された成分核酸分子を物理的に組み立てることによって、識別子核酸分子が形成され、前記Ｍ個の選択された成分核酸分子の各々が、Ｍ個の異なる層に分離された別個の成分核酸分子のセットから選択される、項目１３８～１６４のいずれかに記載の方法。

【図面】

【図１】

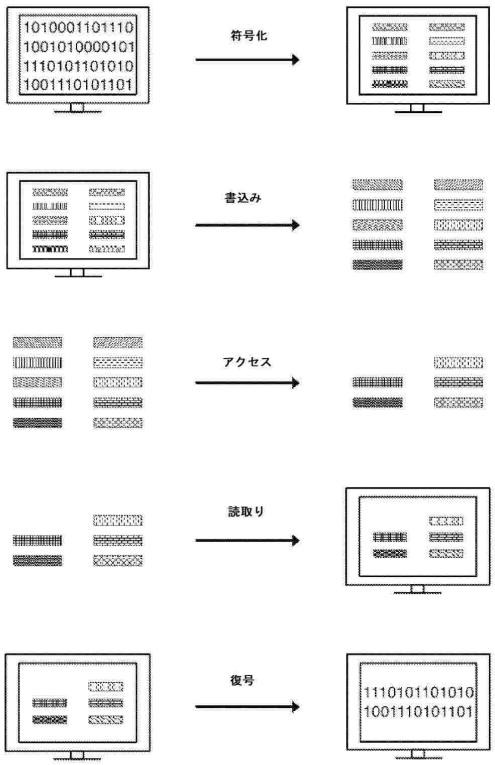



FIG. 1

【図２】

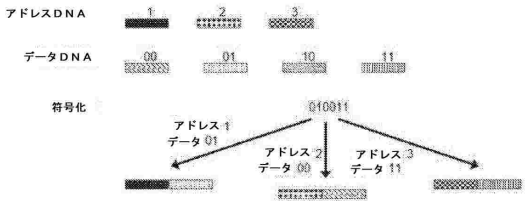



FIG. 2A

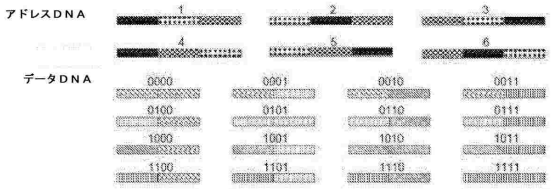



FIG. 2B

【図３】

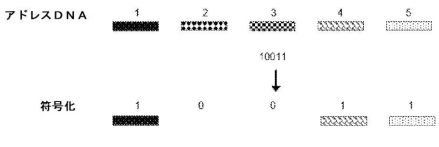



FIG. 3A

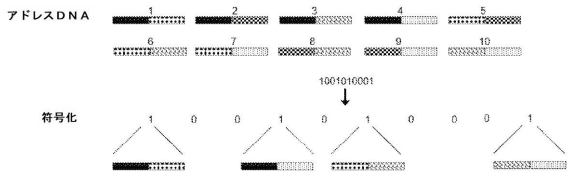



FIG. 3B

【図４】

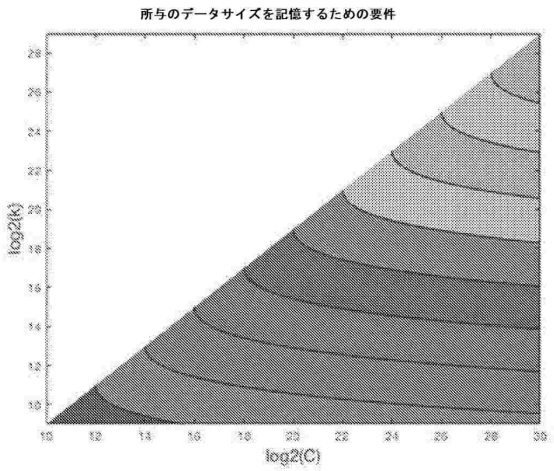



FIG. 4

【図５】

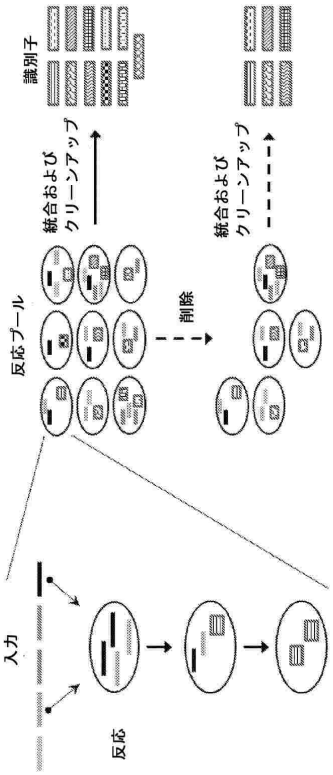



FIG. 5

【図６】

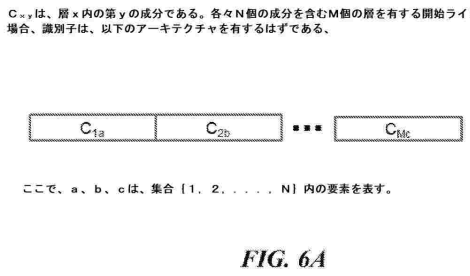



FIG. 6A

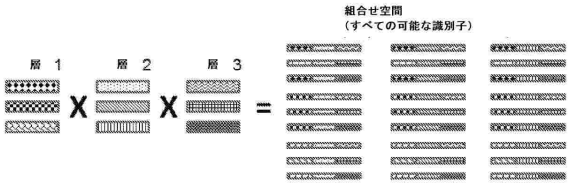



FIG. 6B

【図７】

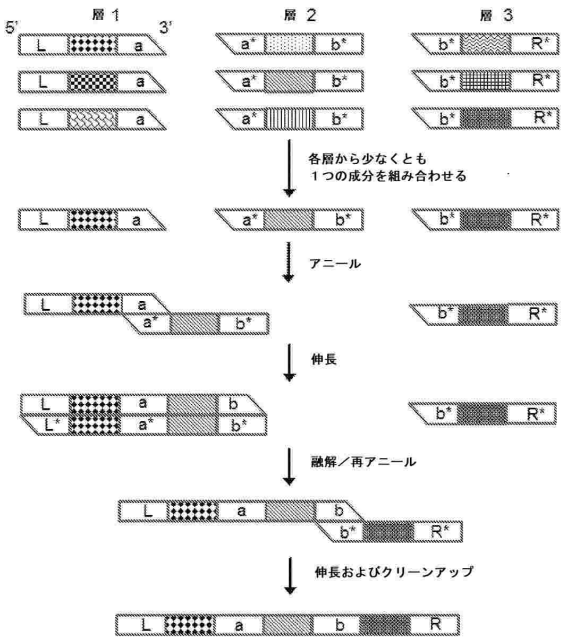



FIG. 7

【図８】

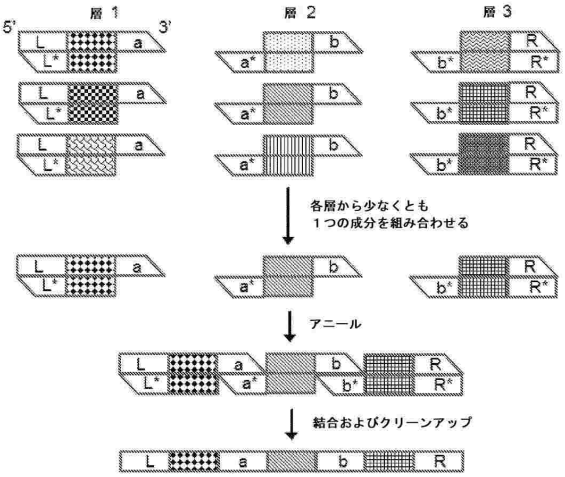



FIG. 8

【図９】

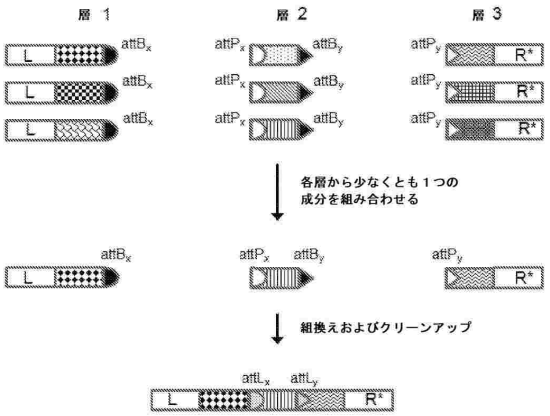



FIG. 9

【図１０Ａ】

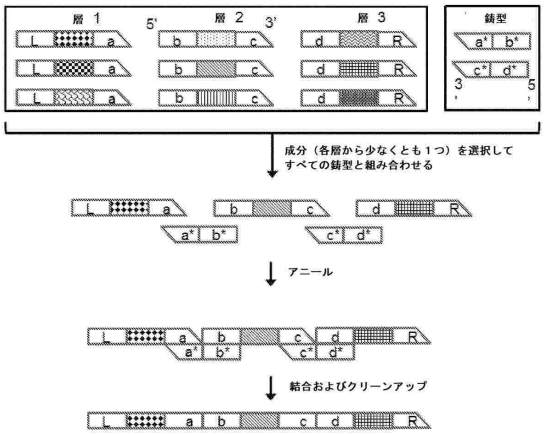



FIG. 10A

【図１０Ｂ】

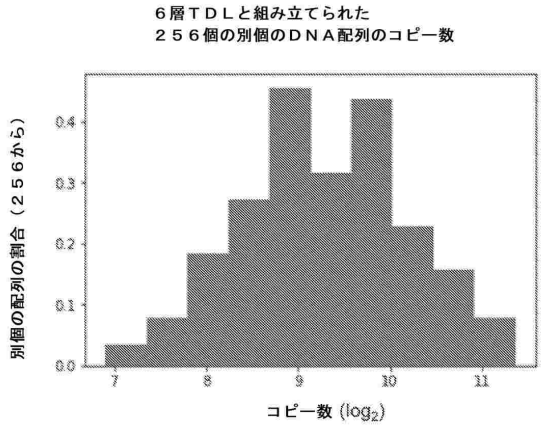



FIG. 10B

【図１１】

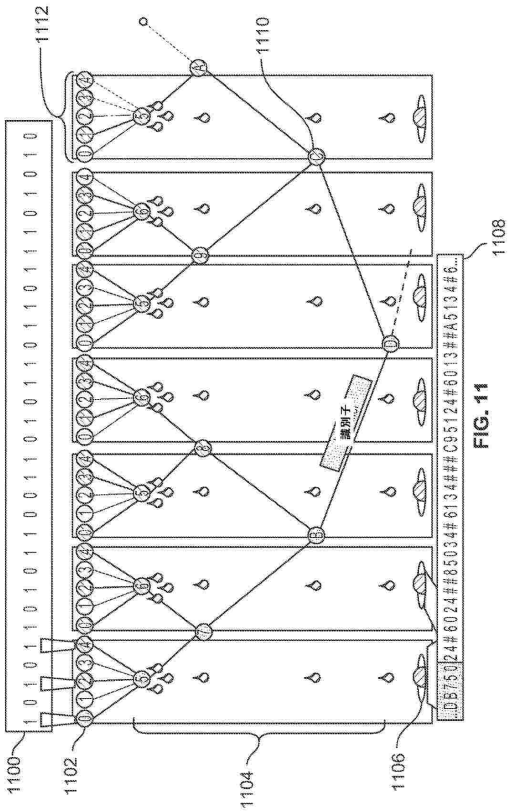



FIG. 11

【図１２】

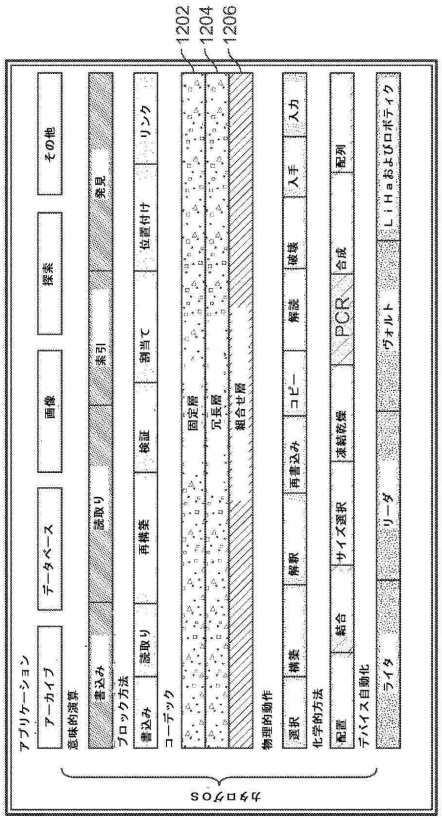



FIG. 12

【図１３】

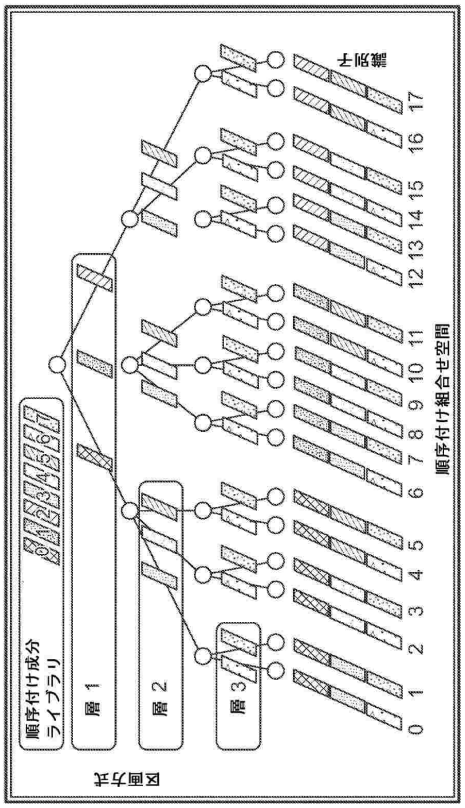



FIG. 13

【図１４】

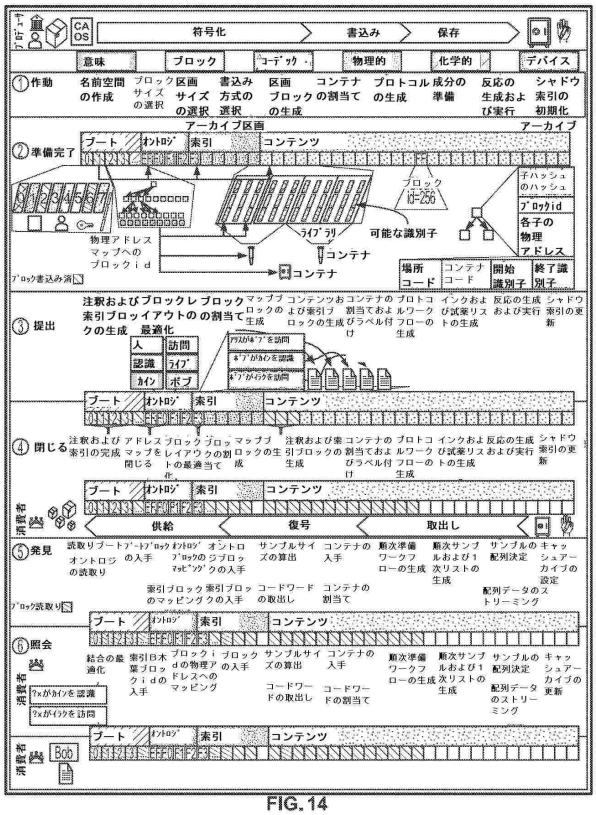



FIG. 14

【図１５】

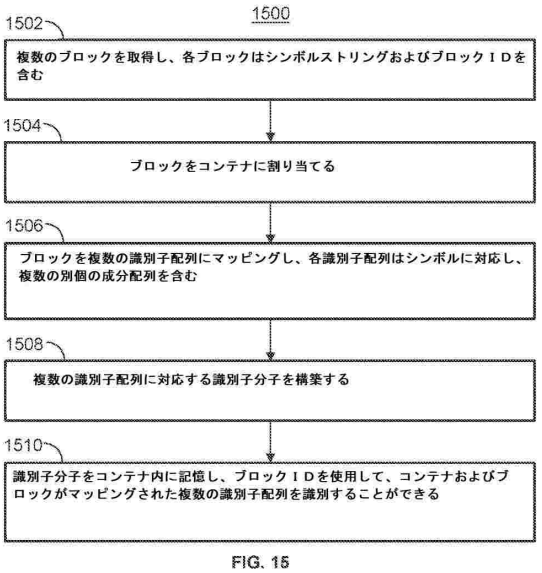
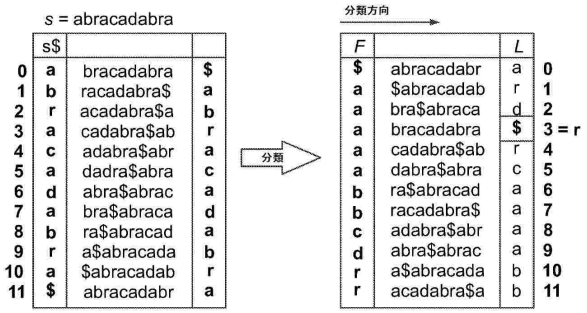



FIG. 15

【図１６】




| | s$ | | |
|---|---|---|---|
| 0 | a | bracadabra | $ |
| 1 | b | racadabra$ | a |
| 2 | r | acadabra$a | b |
| 3 | a | cadabra$ab | r |
| 4 | c | adabra$abr | a |
| 5 | a | dadra$abra | c |
| 6 | d | abra$abrac | a |
| 7 | a | bra$abraca | d |
| 8 | b | ra$abracad | a |
| 9 | r | a$abracada | b |
| 10 | a | $abracadab | r |
| 11 | $ | abracadabr | a |

| | F | | L | |
|---|---|---|---|---|
| | $ | abracadabr | a | 0 |
| | a | $abracadab | r | 1 |
| | a | bra$abraca | d | 2 |
| | a | bracadabra | $ | 3 = r |
| | a | cadabra$ab | r | 4 |
| | a | dabra$abra | c | 5 |
| | b | ra$abracad | a | 6 |
| | b | racadabra$ | a | 7 |
| | c | adabra$abr | a | 8 |
| | d | abra$abrac | a | 9 |
| | r | a$abracada | b | 10 |
| | r | acadabra$a | b | 11 |

FIG. 16

【図１７】

| 接尾辞 | 接尾辞位置 | 分類された接尾辞 | 接尾辞位置 | M′ | L |
|---|---|---|---|---|---|
| abracadabra$ | 0 | $ | 11 | $abracadabra | a |
| bracadabra$ | 1 | a$ | 10 | a$abracadabr | r |
| racadabra$ | 2 | abra$ | 7 | abra$abracad | d |
| acadabra$ | 3 | abracadabra$ | 0 | abracadabra$ | $ |
| cadabra$ | 4 | acadabra$ | 3 | acadabra$abr | r |
| adabra$ | 5 | adabra$ | 5 | adabra$abrc | c |
| dabra$ | 6 | bra$ | 8 | bra$abracada | a |
| abra$ | 7 | bracadabra$ | 1 | bracadabra$a | a |
| bra$ | 8 | cadabra$ | 4 | cadabra$abra | a |
| ra$ | 9 | dabra$ | 6 | dabra$abraca | a |
| a$ | 10 | ra$ | 9 | ra$abracadab | b |
| $ | 11 | racadabra$ | 2 | racadabra$ab | b |

FIG. 17

【図１８】

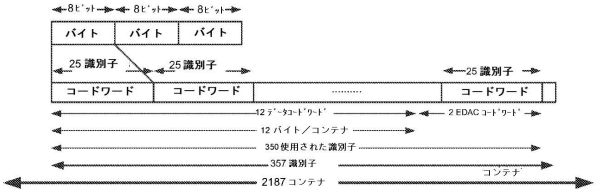



FIG. 18

【図１９】

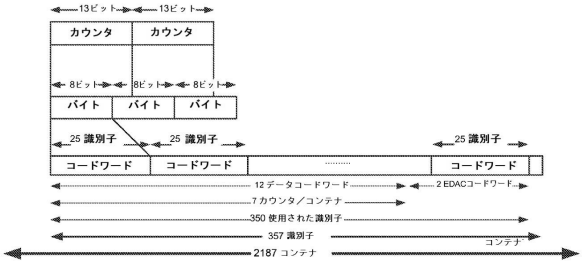



FIG. 19

【図２０】

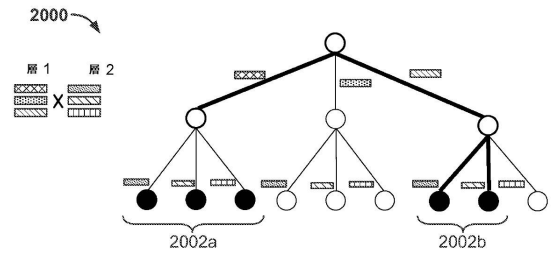



FIG. 20

【図２１】

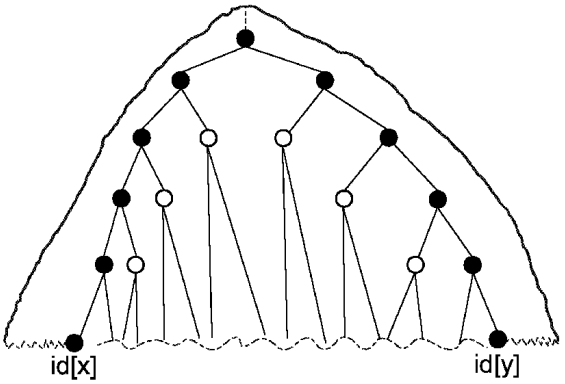



FIG. 21

【図２２】

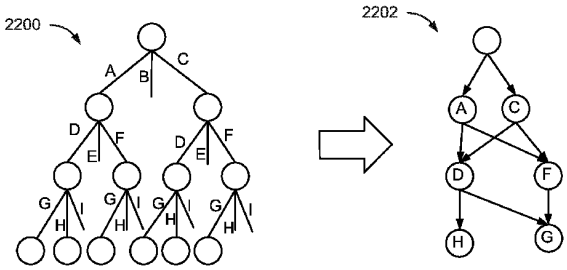



FIG. 22

【図２３】

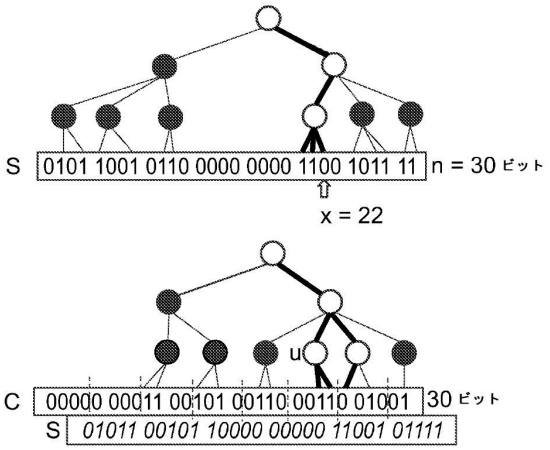



FIG. 23

【図２４】

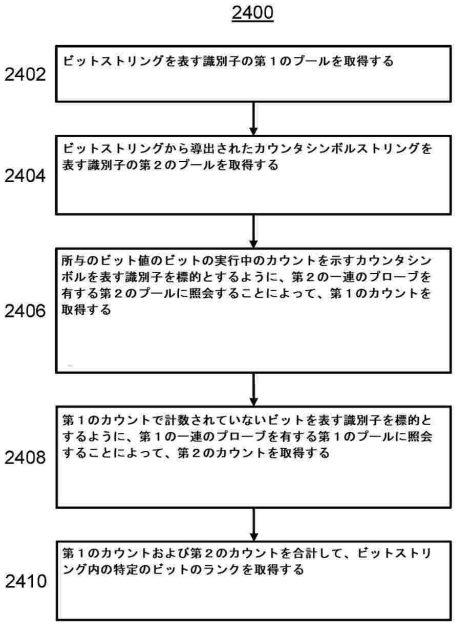



FIG. 24

【図２５】

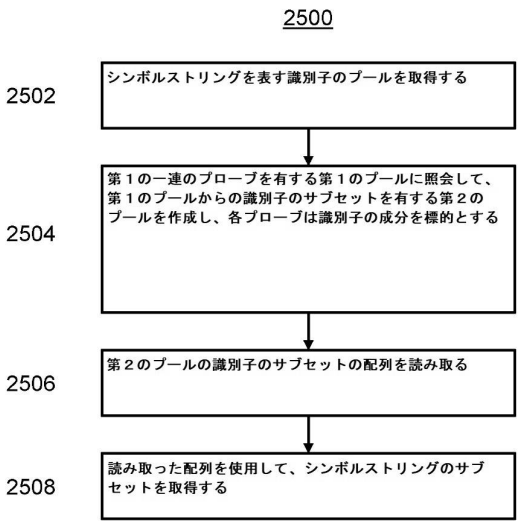



FIG. 25

【図２６】

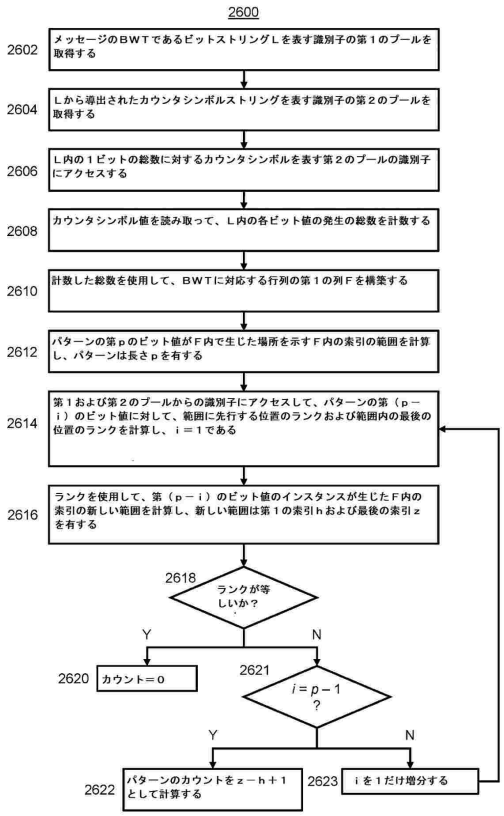



FIG. 26

【図２７】

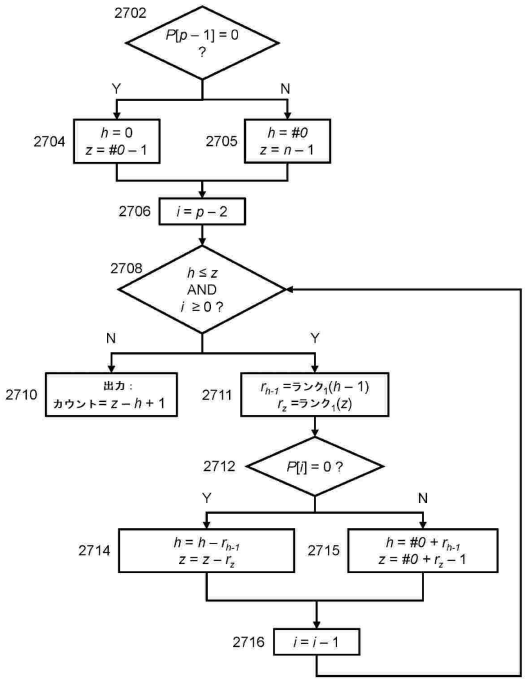



FIG. 27

【図２８】

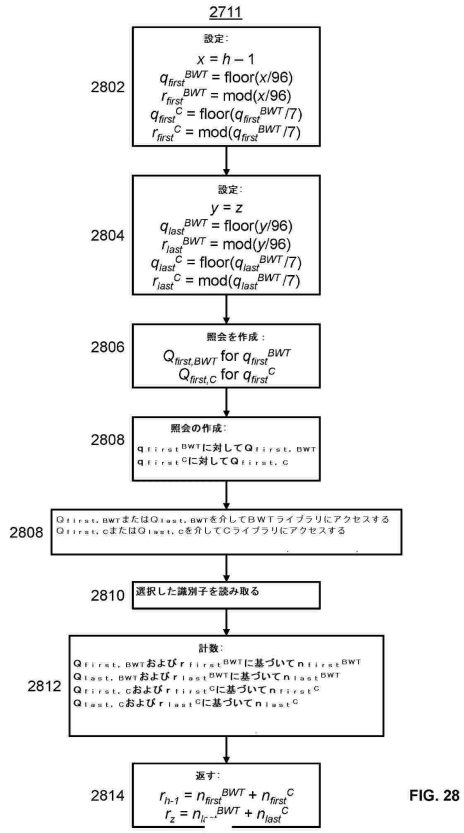



FIG. 28

【図２９】

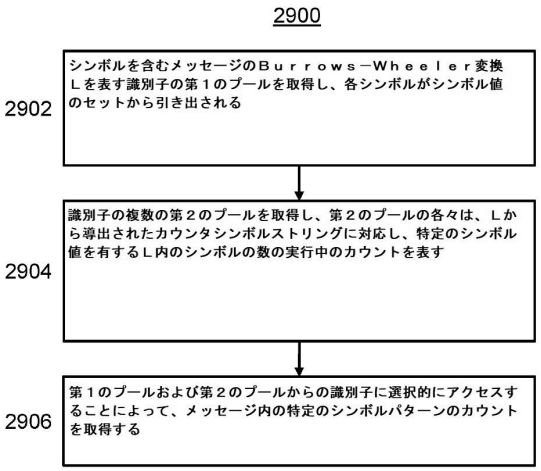



FIG. 29

【図３０】

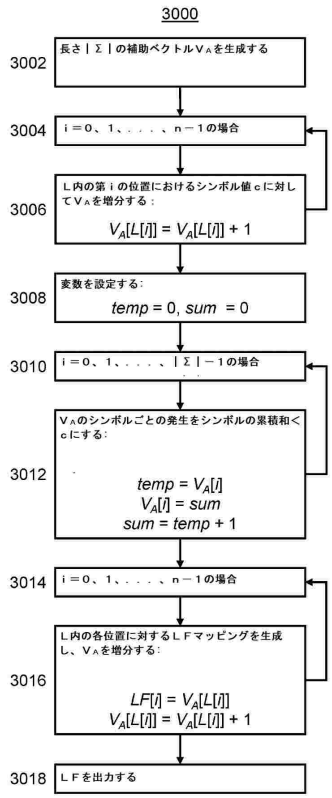



FIG. 30

【図３１】

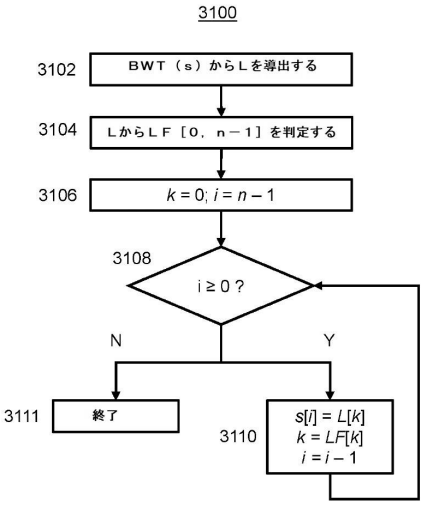



FIG. 31

【図３２】

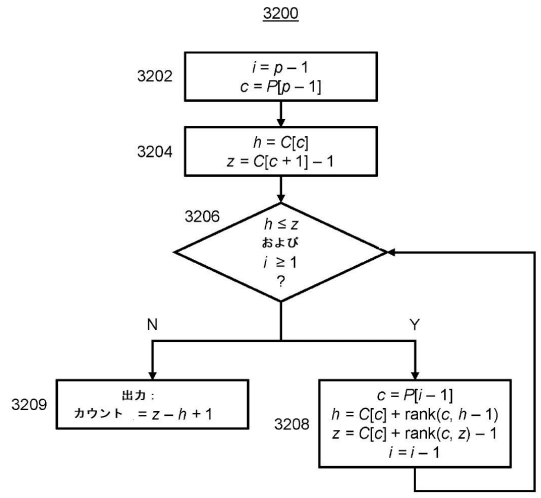



FIG. 32

【図３３】

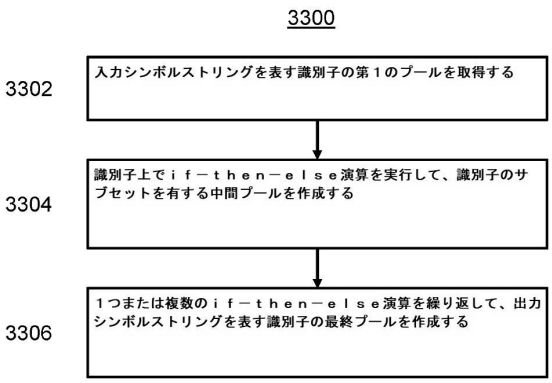



FIG. 33

フロントページの続き

(33)優先権主張国・地域又は機関
米国(US)
(31)優先権主張番号 62/890,243
(32)優先日 令和1年8月22日(2019.8.22)
(33)優先権主張国・地域又は機関
米国(US)
アメリカ合衆国 マサチューセッツ ０２１２９， チャールズタウン， メイン ストリート ５２９， スイート １２７， カタログ テクノロジーズ， インコーポレイテッド 気付
(72)発明者 バティア， スワプニル
アメリカ合衆国 マサチューセッツ ０２１２９， チャールズタウン， メイン ストリート ５２９， スイート １２７， カタログ テクノロジーズ， インコーポレイテッド 気付
(72)発明者 フェラジナ， パオロ
アメリカ合衆国 マサチューセッツ ０２１２９， チャールズタウン， メイン ストリート ５２９， スイート １２７， カタログ テクノロジーズ， インコーポレイテッド 気付
審査官 岡北 有平
(56)参考文献 米国特許出願公開第２０１８／０１３７４１８（ＵＳ，Ａ１）
国際公開第２００４／００９８４４（WO，Ａ１）
特開２００９－２４４９９６（ＪＰ，Ａ）
(58)調査した分野 (Int.Cl.，ＤＢ名)
Ｇ１６Ｂ ５／００ － ９９／００